

OPTIMASI ASUPAN MAKANAN HARIAN IBU HAMIL PENDERITA HIPERTENSI MENGGUNAKAN ALGORITME GENETIKA

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Novirra Dwi Asri
NIM: 145150201111099



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

OPTIMASI ASUPAN MAKANAN HARIAN IBU HAMIL PENDERITA HIPERTENSI MENGUNAKAN ALGORITME GENETIKA

SKRIPSI


Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Novirra Dwi Asri
NIM: 145150201111099

Skrripsi ini telah diuji dan dinyatakan lulus pada
18 Januari 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II


Imam Cholissodin, S.Si, M.Kom
NIK: 201201 850719 1 001


Dian Eka Ratnawati, S.Si, M.Kom
NIP: 19730619 200212 2 001

Mengetahui
Ketua Jurusan Teknik Informatika




Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 18 Januari 2018



Novirra Dwi Asri

NIM: 145150201111099

KATA PENGANTAR

Puji syukur atas kehadiran Allah SWT karena atas segala rahmat dan hidayah-Nya, penulis dapat menyelesaikan skripsi yang berjudul “Optimasi Asupan Makanan Harian Ibu Hamil Penderita Hipertensi Menggunakan Algoritme Genetika”. Skripsi ini dibuat sebagai tugas akhir untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer pada Fakultas Ilmu Komputer Universitas Brawijaya.

Dalam penyusunan skripsi ini juga tak lepas dari bantuan semua pihak. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada:

1. Bapak Imam Cholissodin, S.Si, M.Kom selaku dosen pembimbing I atas segala waktu yang diluangkan dalam memberikan bimbingan serta kritik dan saran sehingga skripsi ini dapat terselesaikan.
2. Ibu Dian Eka Ratnawati, S.Si, M.Kom selaku dosen pembimbing II atas segala waktu yang diluangkan dalam memberikan bimbingan serta kritik dan saran sehingga skripsi ini dapat terselesaikan.
3. Ibu Nawangsari L. selaku ahli gizi dalam skripsi ini atas bimbingan dan informasi yang diberikan kepada penulis.
4. Bapak Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya.
5. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D selaku Ketua Jurusan Teknik Informatika, Fakultas Ilmu Komputer Universitas Brawijaya.
6. Bapak Agus Wahyu Widodo, S.T., M.Cs. selaku Ketua Program Studi Teknik Informatika, Fakultas Ilmu Komputer Universitas Brawijaya.
7. Seluruh bapak ibu dosen dan staff Fakultas Ilmu Komputer atas segala bantuan dan ilmu yang diajarkan kepada penulis
8. Orang tua penulis dan kakak tercinta atas segala perhatian, kasih sayang, dukungan, motivasi, doa dan pengorbanan baik materil maupun moril.
9. Vania Nuraini, Krishnanti Dewi, Revinda Bertananda, Irma Ramadanti, Yulia Kurniawati, Cusen Mosabeth dan Aditya Chandra atas segala dukungan, motivasi, doa, candaan, serta saran yang diberikan kepada penulis.
10. Rekan-rekan Fakultas Ilmu Komputer atas segala bantuan yang diberikan kepada penulis selama berada di Fakultas Ilmu Komputer

Penulis menyadari skripsi ini masih banyak kekurangan dan jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat bermanfaat bagi semua pihak.

Malang, 18 Januari 2018

Penulis
virra.novirra@gmail.com

ABSTRAK

Novirra Dwi Asri. 2017. Optimasi Asupan Makanan Harian Ibu Hamil Penderita Hipertensi Menggunakan Algoritme Genetika.

Pembimbing: Imam Cholissodin, S.Si, M.Kom dan Dian Eka Ratnawati, S.Si, M.Kom.

Hipertensi merupakan salah satu dari penyebab utama angka kematian ibu hamil di Indonesia. Hipertensi adalah keadaan naiknya tekanan darah yang melebihi batas normal yaitu $\geq 140/90$ mmHg. Naiknya tekanan darah dapat dipicu oleh pemilihan asupan makanan yang kurang tepat seperti mengandung kadar lemak berlebihan ataupun mengandung kadar gula berlebihan. Kondisi ini jika dibiarkan dapat memberikan resiko yang tinggi kepada ibu hamil dan janin dalam kandungannya. Dibutuhkan ketelitian dalam menyusun menu makanan bagi ibu hamil penderita Hipertensi. Salah satu metode yang dapat diterapkan dalam penyelesaian permasalahan ini adalah Algoritme Genetika. Algoritme Genetika merupakan metode *heuristic* yang menggunakan aturan-aturan untuk mendapatkan solusi terbaik pada suatu permasalahan. Algoritme Genetika banyak dimanfaatkan dalam melakukan optimasi penyusunan menu makanan. Dalam penyusunan kromosom digunakan representasi bilangan integer, metode *crossover* yang digunakan adalah *extended intermediate crossover*, metode mutasi yang digunakan adalah *random mutation* dan proses seleksi menggunakan metode seleksi *elitism*. Hasil yang direkomendasikan adalah menu makanan selama beberapa hari. Pada hasil pengujian dalam permasalahan ini didapatkan ukuran generasi optimal sebesar 240, ukuran populasi optimal sebesar 90 dan kombinasi nilai *cr* dan *mr* optimal adalah 0,6 dan 0,5. Berdasarkan hasil tersebut sistem mampu memberikan rekomendasi makanan dengan mempertimbangkan penyakit Hipertensi.

Kata kunci: ibu hami, Hipertensi, makanan, optimasi, Algoritme Genetika

ABSTRACT

Novirra Dwi Asri. 2017. Optimization of Daily Food for Hypertension Pregnant Women Using Genetic Algorithm.

Adviser: Imam Cholissodin, S.Si, M.Kom and Dian Eka Ratnawati, S.Si, M.Kom.

Hypertension is one of the two main causes the death of pregnant women in Indonesia. Hypertension is a situation of elevated blood pressure beyond normal limits $\geq 140 / 90$ mmHg. The increased of blood pressure can be caused by the selection of foods that are at risk such as containing more of fat or containing more of sugar. If this condition is not monitored, it can provide a high risk to pregnant women and their fetus. This condition requires precision in arrange the food menu for pregnant women with hypertension. The one method that can be applied in solving this problem is Genetic Algorithm. Genetic algorithm is a heuristic method that uses rules to get the best solution for the problem. Genetic Algorithm is widely used for food menu optimization. In the representation of chromosomes used the representation of integer numbers, crossover method that used is extended intermediate crossover, mutation method that used is random mutation and selection method using elitism selection. The recommendation result is a food menu for seven days. Based on test results in this problem, the optimal generation size is 240, the optimal population size is 90 and the combination of cr and mr optimal is 0.6 and 0.5. From these results the system is able to provide food recommendations that meet daily nutritional needs.

Keywords: pregnant women, Hypertension, food, optimization, Genetic Algorithm

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xii
DAFTAR PERSAMAAN.....	xiii
DAFTAR KODE PROGRAM	xiv
DAFTAR LAMPIRAN	xv
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah	2
1.3 Tujuan	2
1.4 Manfaat.....	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian pustaka	5
2.2 Kehamilan	7
2.3 Ibu hamil penderita Hipertensi.....	7
2.4 Kebutuhan gizi ibu hamil	7
2.4.1 Kebutuhan energi.....	7
2.4.2 Karbohidrat	10
2.4.3 Protein.....	10
2.4.4 Lemak	10
2.4.5 Natrium	11
2.5 Algoritme genetika.....	11
2.5.1 Inisialisasi	12

2.5.2 Reproduksi	12
2.5.3 Evaluasi.....	13
2.5.4 Seleksi.....	15
BAB 3 METODOLOGI	16
3.1 Studi pustaka	16
3.2 Pengumpulan data.....	17
3.3 Analisis kebutuhan sistem	17
3.4 Perancangan sistem	17
3.5 Implementasi sistem.....	18
3.6 Pengujian sistem	18
3.7 Pengambilan kesimpulan.....	18
BAB 4 PERANCANGAN.....	19
4.1 Formulasi permasalahan.....	19
4.2 Siklus penyelesaian masalah.....	19
4.3 Tahapan algoritme genetika	20
4.3.1 Inisialisasi populasi awal	21
4.3.2 <i>Crossover</i>	22
4.3.3 Mutasi	24
4.3.4 Evaluasi.....	26
4.3.5 Seleksi.....	28
4.4 Contoh perhitungan manual.....	29
4.4.1 Contoh permasalahan	29
4.4.2 Inisialisasi parameter algoritme genetika	31
4.4.3 Representasi kromosom	31
4.4.4 Inisialisasi populasi awal	34
4.4.5 <i>Crossover</i>	34
4.4.6 Mutasi	35
4.4.7 Evaluasi.....	36
4.4.8 Seleksi.....	38
4.5 Perancangan antarmuka	39
4.5.1 Perancangan antarmuka halaman utama.....	39
4.5.2 Perancangan antarmuka halaman proses algoritme genetika ...	40

4.5.3 Perancangan antarmuka halaman hasil optimasi	40
4.5.4 Perancangan antarmuka halaman daftar makanan	41
4.6 Perancangan pengujian	41
4.6.1 Pengujian ukuran generasi.....	42
4.6.2 Pengujian ukuran populasi.....	42
4.6.3 Pengujian kombinasi <i>crossover rate</i> dan <i>mutation rate</i>	43
BAB 5 IMPLEMENTASI	44
5.1 Spesifikasi sistem	44
5.1.1 Spesifikasi perangkat keras	44
5.1.2 Spesifikasi perangkat lunak.....	44
5.2 Implementasi program	44
5.2.1 Implementasi proses inialisasi populasi awal.....	45
5.2.2 Implementasi proses <i>crossover</i>	46
5.2.3 Implementasi proses mutasi.....	47
5.2.4 Implementasi perhitungan penalti	47
5.2.5 Implementasi proses perhitungan <i>fitness</i>	50
5.2.6 Implementasi proses seleksi	51
5.3 Implementasi antarmuka.....	52
5.3.1 Implementasi antarmuka halaman utama.....	52
5.3.2 Implementasi antarmuka halaman proses algoritme genetika ..	53
5.3.3 Implementasi antarmuka halaman hasil menu makanan	55
5.3.4 Implementasi antarmuka halaman daftar makanan	55
BAB 6 PENGUJIAN DAN PEMBAHASAN.....	56
6.1 Pengujian ukuran generasi	56
6.2 Pengujian ukuran populasi	57
6.3 Pengujian kombinasi <i>crossover rate</i> dan <i>mutation rate</i>	59
6.4 Hasil analisis global	60
BAB 7 PENUTUP	62
7.1 Kesimpulan.....	62
7.2 Saran	62
DAFTAR PUSTAKA.....	64
LAMPIRAN A DATA IBU HAMIL.....	66

LAMPIRAN B DATA MAKANAN.....	67
LAMPIRAN C HASIL PENGUJIAN	73
LAMPIRAN D HASIL REKOMENDASI MAKANAN.....	76



DAFTAR TABEL

Tabel 2.1 Kajian pustaka	6
Tabel 2.2 Klasifikasi Berat Badan Ideal (BBI)	8
Tabel 2.3 Nilai usia	9
Tabel 2.4 Nilai aktivitas fisik	9
Tabel 2.5 Nilai kalori kehamilan	9
Tabel 2.6 Tambahan kebutuhan gizi ibu hamil	10
Tabel 4.1 Sumber karbohidrat	19
Tabel 4.2 Contoh representasi kromosom	32
Tabel 4.3 Konversi kromosom menjadi menu makanan	32
Tabel 4.4 Inisialisasi populasi awal	34
Tabel 4.5 Proses <i>crossover</i>	35
Tabel 4.6 <i>Offspring</i> hasil <i>crossover</i>	35
Tabel 4.7 Induk mutasi	35
Tabel 4.8 <i>Offspring</i> hasil mutasi	36
Tabel 4.9 Populasi baru	36
Tabel 4.10 Jumlah kandungan gizi makanan sistem	37
Tabel 4.11 Perhitungan variasi	38
Tabel 4.12 Hasil perhitungan <i>fitness</i>	38
Tabel 4.13 Hasil seleksi	39
Tabel 4.14 Hasil perhitungan <i>fitness</i> generasi ke-2	39
Tabel 4.15 Pengujian ukuran generasi	42
Tabel 4.16 Pengujian ukuran populasi	42
Tabel 4.17 Pengujian <i>cr</i> dan <i>mr</i>	43
Tabel 5.1 Spesifikasi perangkat keras	44
Tabel 5.2 Spesifikasi perangkat lunak	44
Tabel 6.1 Hasil pengujian ukuran generasi	56
Tabel 6.2 Hasil pengujian ukuran populasi	57
Tabel 6.3 Hasil pengujian kombinasi <i>cr</i> dan <i>mr</i>	59
Tabel 6.4 Hasil perhitungan kebutuhan gizi aktual	60
Tabel 6.5 Selisih kebutuhan gizi dan kandungan gizi hasil sistem	60

DAFTAR GAMBAR

Gambar 2.1 Alur Algoritme Genetika.....	12
Gambar 2.2 Contoh <i>extended intermediate crossover</i>	13
Gambar 2.3 Contoh <i>random mutation</i>	13
Gambar 3.1 Diagram alir penelitian	16
Gambar 4.1 Diagram alir algoritme genetika.....	20
Gambar 4.2 Diagram alir inisialisasi populasi awal	21
Gambar 4.3 Diagram alir proses <i>crossover</i>	22
Gambar 4.4 Diagram alir <i>extended intermediate crossover</i>	24
Gambar 4.5 Diagram alir proses mutasi	25
Gambar 4.6 Diagram alir <i>random mutation</i>	26
Gambar 4.7 Diagram alir evaluasi	27
Gambar 4.8 Diagram alir seleksi <i>elitism</i>	29
Gambar 4.9 Perancangan antarmuka halaman utama	40
Gambar 4.10 Perancangan antarmuka halaman proses algoritme genetika	40
Gambar 4.11 Perancangan antarmuka halaman hasil menu makanan.....	41
Gambar 4.12 Perancangan antarmuka halaman daftar makanan	41
Gambar 5.1 Antarmuka halaman utama	52
Gambar 5.2 Antarmuka halaman populasi awal.....	53
Gambar 5.3 Antarmuka halaman hasil <i>crossover</i>	53
Gambar 5.4 Antarmuka halaman hasil mutasi	54
Gambar 5.5 Antarmuka halaman individu terbaik	54
Gambar 5.6 Antarmuka halaman hasil menu makanan	55
Gambar 5.7 Antarmuka halaman daftar makanan	55
Gambar 6.1 Grafik hasil pengujian ukuran generasi.....	57
Gambar 6.2 Grafik hasil pengujian ukuran populasi.....	58
Gambar 6.3 Hasil pengujian kombinasi <i>cr</i> dan <i>mr</i>	59

DAFTAR PERSAMAAN

Persamaan (2.1) 8

Persamaan (2.2) 8

Persamaan (2.3) 8

Persamaan (2.4) 9

Persamaan (2.5) 9

Persamaan (2.6) 10

Persamaan (2.7) 12

Persamaan (2.8) 12

Persamaan (2.9) 13

Persamaan (2.10) 14

Persamaan (2.11) 14

Persamaan (2.12) 14

Persamaan (2.13) 14

Persamaan (2.14) 14

Persamaan (2.15) 14

Persamaan (2.16) 14

Persamaan (2.17) 14



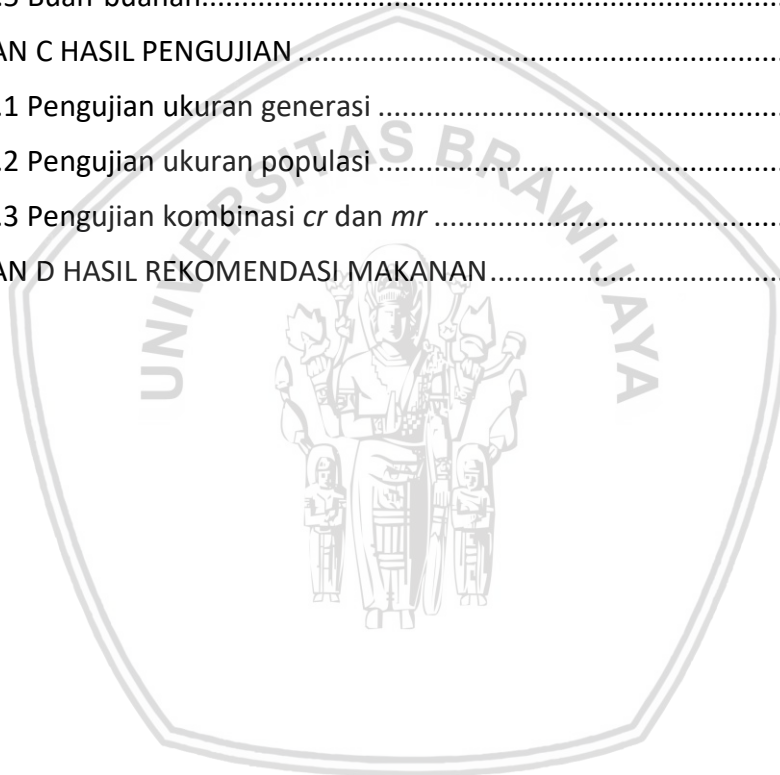
DAFTAR KODE PROGRAM

Kode Program 5.1 Implementasi proses inialisasi populasi awal.....	45
Kode Program 5.2 Implementasi proses <i>crossover</i>	46
Kode Program 5.3 Implementasi proses mutasi	47
Kode Program 5.4 Implementasi proses perhitungan penalti	50
Kode Program 5.5 Implementasi proses perhitungan <i>fitness</i>	51
Kode Program 5.6 Implementasi proses seleksi	52



DAFTAR LAMPIRAN

LAMPIRAN A DATA IBU HAMIL.....	66
LAMPIRAN B DATA MAKANAN.....	67
B.1 Sumber karbohidrat.....	67
B.2 Sumber protein hewani	68
B.3 Sumber protein nabati.....	69
B.4 Sayuran	70
B.5 Buah-buahan.....	71
LAMPIRAN C HASIL PENGUJIAN	73
C.1 Pengujian ukuran generasi	73
C.2 Pengujian ukuran populasi	74
C.3 Pengujian kombinasi <i>cr</i> dan <i>mr</i>	75
LAMPIRAN D HASIL REKOMENDASI MAKANAN	76



BAB 1 PENDAHULUAN

1.1 Latar belakang

Asupan makanan merupakan kebutuhan yang sangat penting bagi manusia untuk menjalankan aktivitasnya sehari-hari. Dengan asupan makanan yang bergizi maka akan menjaga kesehatan badan agar tetap sehat. Berbagai jenis makanan pada dasarnya boleh untuk dikonsumsi oleh manusia, namun manusia seringkali secara berlebihan dalam mengonsumsi suatu jenis makanan, dengan pola makan yang salah tersebut menyebabkan kenaikan berat badan yang berlebihan dan timbulnya berbagai macam penyakit.

Bagi seorang ibu hamil khususnya sangatlah penting asupan makanan harian yang bergizi untuk menjaga kesehatannya dan janin yang ada dalam kandungannya. Semua makanan yang dimakan oleh ibu hamil juga merupakan nutrisi untuk janinnya. Asupan makanan harian bagi ibu hamil menentukan bagaimana perkembangan janin yang ada dalam kandungannya. Penyakit yang diderita ibu hamil pun dapat mempengaruhi jenis makanan yang dapat dikonsumsi, dengan adanya penyakit tersebut maka diperlukan ketelitian dalam memilih jenis makanan agar tetap bergizi dan mengandung nutrisi yang baik untuk janin yang ada dalam kandungannya.

Hipertensi merupakan salah satu penyakit yang dapat mengganggu proses kehamilan ibu hamil. Pertumbuhan dan perkembangan janin dapat terganggu dengan adanya penyakit tersebut. Ibu hamil penderita Hipertensi memiliki resiko tinggi dalam proses persalinannya, kondisi Hipertensi sangat dipengaruhi oleh konsumsi makanan sehari-hari. Pemilihan makanan yang kurang tepat dapat meningkatkan kondisi tekanan darah. Bagi ibu hamil ataupun ahli gizi proses menentukan jenis makanan untuk dikonsumsi merupakan hal yang tidak mudah dan membutuhkan waktu yang lama jika dilakukan secara manual. Oleh karena itu diperlukan sebuah sistem yang dapat membantu ibu hamil dan ahli gizi untuk dapat dengan mudah merencanakan asupan makanan harian yang sesuai dengan mempertimbangkan penyakit yang diderita oleh ibu hamil yaitu Hipertensi.

Berdasarkan laporan World Health Organization (WHO) angka kematian ibu hamil di dunia akibat Hipertensi pada saat persalinan mencapai 536.000 wanita (WHO, 2005). Di Indonesia Hipertensi pada ibu hamil termasuk salah satu dari dua penyebab utama kematian ibu hamil, berdasarkan data Kemenkes, sebesar 28.8% ibu hamil di Indonesia menderita Hipertensi (Humaniora, 2016). Di daerah Jawa Timur, yaitu Kabupaten Tulungagung menurut laporan Dinas Kesehatan Kabupaten Tulungagung mayoritas penyebab kematian ibu melahirkan pada daerah tersebut karena pendarahan yang dipicu oleh kondisi Hipertensi. Angka kematian pada daerah tersebut dapat mencapai lebih dari 50% dari total kejadian kematian ibu hamil (Diskes, 2017).

Berdasarkan data kasus ibu hamil penderita Hipertensi maka sangat diperlukan pengaturan asupan gizi pada ibu hamil penderita Hipertensi. Didalam melakukan penelitian ini maka diperlukan algoritme yang mampu menyelesaikan

permasalahan penyusunan makanan. Algoritme genetika merupakan algoritme yang mampu untuk menyelesaikan masalah penyusunan makanan. Dalam berbagai bidang algoritme genetika telah banyak dimanfaatkan seperti optimasi penugasan mengajar dosen, optimasi distribusi produk, penentuan komposisi pakan ternak dan lain sebagainya (Mahmudy, 2015).

Penelitian sebelumnya yang dilakukan oleh Nurbaiti Wahid pada tahun 2015 yaitu optimasi komposisi makanan untuk penderita kolesterol. Digunakan 100 bahan makanan pada penelitian tersebut dengan panjang kromosom 9 menghasilkan solusi optimal pada populasi berukuran 100 dengan rata-rata nilai *fitness* sebesar 0.1862463, ukuran generasi 90, nilai $cr = 0.7$ dan $mr = 0.3$ mendapatkan hasil rata-rata nilai *fitness* sebesar 0.18575847. Penelitian lain juga pernah dilakukan oleh Maryamah pada tahun 2017 yaitu optimasi komposisi makanan penderita diabetes dan komplikasinya dengan harga terjangkau. Hasil pada penelitian tersebut adalah komposisi makanan dalam satu hari yang terdiri dari makan pagi, makan siang, makan malam. Komposisi makanan yang dihasilkan telah sesuai dengan kebutuhan gizi serta dengan harga terjangkau.

Berdasarkan penelitian yang telah dilakukan sebelumnya menunjukkan algoritme genetika dapat menyelesaikan masalah-masalah yang berkaitan dengan penyusunan menu makanan, maka dalam penelitian ini yang berjudul "Optimasi Asupan Makanan Harian Ibu Hamil Penderita Hipertensi Menggunakan Algoritme Genetika" penulis menggunakan algoritme genetika dalam menyelesaikan masalah dengan melibatkan pakar yaitu seorang ahli gizi memberikan informasi mengenai menu makanan bergizi yang disarankan untuk ibu hamil dengan penyakit Hipertensi.

1.2 Rumusan masalah

Berdasarkan penjelasan pada latar belakang maka dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimana implementasi algoritme genetika dalam optimasi asupan makanan harian ibu hamil penderita Hipertensi?
2. Bagaimana kualitas hasil keluaran asupan makanan harian ibu hamil penderita Hipertensi dengan menggunakan algoritme genetika?

1.3 Tujuan

1. Mengimplementasikan algoritme genetika dalam optimasi asupan makanan harian ibu hamil penderita Hipertensi.
2. Mengetahui kualitas hasil keluaran sistem dalam menentukan asupan makanan harian ibu hamil penderita Hipertensi dengan menerapkan algoritme genetika.

1.4 Manfaat

Manfaat yang ingin diperoleh dengan adanya penelitian ini antara lain:

1. Membantu ibu hamil penderita Hipertensi dalam menentukan makanan yang dikonsumsi dengan mempertimbangkan penyakit Hipertensi.
2. Memudahkan pekerjaan ahli gizi agar lebih mudah dan cepat dalam menentukan makanan harian yang sesuai untuk ibu hamil dengan Hipertensi.

1.5 Batasan masalah

1. Data yang digunakan dalam penelitian adalah data ibu hamil penderita Hipertensi yang didapat dari Bidan Praktik Swasta Mojokerto.
2. Data makanan yang digunakan dalam penelitian ini adalah daftar makanan olahan beserta kandungan gizinya.
3. *Output* yang akan dihasilkan berdasarkan data yang digunakan adalah daftar makanan dalam beberapa hari yang terdiri dari makan pagi, makan siang, dan makan malam.
4. *Output* asupan makanan yang dihasilkan sesuai dengan kebutuhan kalori harian, usia kehamilan dan mempertimbangkan penyakit Hipertensi yang dialami oleh ibu hamil.

1.6 Sistematika pembahasan

BAB 1 PENDAHULUAN

Bab ini akan menjelaskan mengenai latar belakang, rumusan masalah, tujuan, manfaat dan sistematika pembahasan yang ada dalam penelitian ini.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini akan menjelaskan mengenai kajian pustaka yang berisi penjelasan mengenai penelitian-penelitian sebelumnya dan dasar teori yang berkaitan dengan penelitian yang akan dilakukan.

BAB 3 METODOLOGI

Bab ini akan menjelaskan langkah-langkah yang dilakukan dalam penelitian antara lain studi pustaka, pengumpulan data, analisis kebutuhan sistem, perancangan sistem, implementasi sistem, pengujian sistem dan pengambilan kesimpulan.

BAB 4 PERANCANGAN

Bab ini akan menjelaskan proses perancangan sistem yang akan dibangun. Proses perancangan tersebut meliputi formulasi permasalahan, siklus penyelesaian masalah, tahapan algoritme genetika, perhitungan manual, perancangan antarmuka, dan perancangan pengujian.

BAB 5 IMPLEMENTASI

Bab ini akan menjelaskan spesifikasi sistem yang digunakan, implementasi program, dan implementasi antarmuka sistem.

BAB 6 PENGUJIAN DAN PEMBAHASAN

Bab ini akan menjelaskan mengenai hasil pengujian dan pembahasan hasil pengujian sistem.

BAB 7 PENUTUP

Bab ini akan menjelaskan kesimpulan dari hasil akhir dan seluruh proses yang telah dilakukan dalam membangun sistem, serta saran agar penelitian ini dapat dikembangkan lebih baik lagi pada penelitian selanjutnya.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian pustaka

Kajian pustaka pada penelitian ini akan menjelaskan penelitian-penelitian yang sudah pernah dilakukan sebelumnya. Pada penelitian pertama dilakukan oleh Fitria Dwi Kusumaningsih pada tahun 2016. Pada penelitian ini melakukan optimasi kombinasi bahan makanan untuk anggota keluarga. Hasil yang diberikan adalah rekomendasi bahan makanan selama 7 hari dengan biaya minimum. Hasil rekomendasi dengan algoritma genetika tersebut berhasil menghemat biaya rata-rata sebesar 15,49% dengan parameter algoritma genetika yang tepat adalah jumlah generasi 80, populasi 80 dan nilai kombinasi cr dan mr sebesar 0.5:0.5.

Pada penelitian yang kedua dilakukan oleh Muhammad Shafaat pada tahun 2017. Penelitian tersebut melakukan optimasi pada komposisi makanan diet bagi penderita Hipertensi. Hasil yang diberikan adalah rekomendasi makanan untuk beberapa hari yang terdiri dari makan pagi, makan siang, dan makan malam dengan biaya minimal. Pada penelitian tersebut hasil optimal diperoleh pada ukuran populasi 80 dengan nilai *fitness* 1100,495433, jumlah generasi 300 dengan nilai *fitness* sebesar 1190,022286 serta nilai $cr=0,1$, $mr=0,9$ dan nilai *fitness* sebesar 1150,466927.

Penelitian ketiga dilakukan oleh Maryamah pada tahun 2017. Penelitian tersebut melakukan optimasi komposisi makanan pada penderita diabetes melitus dan komplikasinya. Hasil yang diberikan adalah susunan makanan diet bergizi dengan harga terjangkau. Dalam penelitian tersebut hasil yang didapatkan adalah individu dalam populasi optimal berjumlah 250 individu, jumlah generasi 145, nilai cr dan mr optimal adalah 0,7 dan 0,3.

Penelitian keempat dilakukan oleh Nurbaiti Wahid pada tahun 2015. Penelitian tersebut melakukan optimasi pada komposisi makanan untuk penderita Kolesterol. Hasil yang diberikan oleh penelitian tersebut membuktikan bahwa parameter Algoritma Genetika mempengaruhi hasil nilai *fitness*. Jumlah generasi yang semakin besar dibandingkan dengan populasi akan menyebabkan konvergensi. Penelitian tersebut mendapatkan hasil optimal pada generasi 90 dengan nilai rata-rata *fitness* 0,1838946, populasi yang optimal adalah 100 dengan nilai rata-rata *fitness* 0,1862463, kombinasi nilai cr dan mr adalah 0,7 dan 0,3.

Penelitian kelima dilakukan oleh Aprilia Nur Fauziyah. Hasil keluaran tersebut adalah komposisi makanan dengan memperhatikan kadar garam dan biaya pada makanan untuk penderita Hipertensi menggunakan Algoritme *Variable Neighborhood Search (VNS)*. Data makanan yang digunakan berjumlah 103 data. Pada hasil pengujian menunjukkan rata-rata *fitness* tertinggi sebesar 0,516848 dengan nilai $K_{max} = 13$ dan rata-rata *fitness* terbesar yaitu 0,524301 dengan jumlah iterasi local search 2000. Hasil kandungan gizi komposisi makanan pada penelitian tersebut mendekati kebutuhan penderita Hipertensi dengan memperhatikan kadar garam natrium dan biaya minimal dalam satu hari.

Penelitian keenam dilakukan oleh Ayu Puspo Sari pada tahun 2014. Penelitian tersebut melakukan optimasi gizi untuk ibu hamil dengan hasil keluaran yaitu makan pagi, makan siang, dan makan malam. Namun, saat ini telah banyak ditemukan kasus ibu hamil yang menderita penyakit Hipertensi, sehingga penelitian tersebut kurang cocok digunakan dalam menyusun makanan untuk ibu hamil dengan penyakit Hipertensi. Oleh karena itu sangat penting untuk melakukan penelitian optimasi asupan makanan harian bagi ibu hamil penderita Hipertensi.

Berdasarkan penelitian yang sudah ada sebelumnya maka penulis mengusulkan sebuah penelitian dengan menggunakan algoritme genetika dalam optimasi asupan makanan harian ibu hamil penderita Hipertensi. *Input* yang akan digunakan antara lain usia ibu hamil, usia kehamilan, berat badan, tinggi badan, aktivitas dan tekanan darah. Sistem akan menghitung besarnya energi yang dibutuhkan oleh ibu hamil dan memberikan hasil keluaran asupan makanan harian yang sesuai selama beberapa hari. Kajian pustaka yang berkaitan dengan penelitian dijelaskan pada Tabel 2.1.

Tabel 2.1 Kajian pustaka

No.	Pustaka	Objek	Metode	Output
1	Fitria Dwi Kusumaningsih, Imam Cholissodin, Budi Darma Setiawan, 2016	Bahan makanan	Algoritme Genetika	Rekomendasi kombinasi bahan makanan untuk anggota keluarga selama tujuh hari berdasarkan harga minimum.
2	Muhammad Shafaat, Imam Cholissodin, Edy Santoso, 2017	Komposisi makanan penderita Hipertensi	Algoritme Genetika	Rekomendasi komposisi makanan untuk beberapa hari
3	Maryamah, Rekyan Regarsari Mardi Putri, Satrio Agung Wicaksono, 2017	Komposisi makanan penderita diabetes mellitus dan komplikasinya	Algoritme Genetika	Komposisi makanan yang memenuhi gizi dengan harga terjangkau untuk penderita diabetes dan komplikasinya.
4	Nurbaiti Wahid, Wayan Firdaus Mahmudy, 2015	Komposisi makanan penderita kolesterol	Algoritme Genetika	Rekomendasi proporsi luas lahan yang optimal untuk digunakan sebagai lahan pertanian.
5	Aprilia Nur Fauziyah, Wayan Firdaus Mahmudy, 2017	Komposisi makanan penderita Hipertensi	<i>Variable Neighborhood Search</i>	Rekomendasi komposisi makanan rendah natrium untuk penderita Hipertensi dengan biaya minimal dalam satu hari
6	Ayu Puspo Sari, Wayan Firdaus Mahmudy, Candra Dewi, 2014	Gizi ibu hamil	Algoritme genetika	Rekomendasi makanan dalam satu hari

Tabel 2.2 Kajian pustaka (lanjutan)

No.	Pustaka	Objek	Metode	Output
7	Penulis	Susunan makanan ibu hamil penderita Hipertensi	Algoritme Genetika	Rekomendasi susunan makanan untuk ibu hamil penderita Hipertensi dalam beberapa hari terdiri dari makan pagi, makan siang, makan malam.

2.2 Kehamilan

Kehamilan bagi seorang ibu menyebabkan peningkatan kebutuhan energi dan zat gizi lainnya sebagai akibat dari meningkatnya metabolisme energi (Ariani, 2017). Adanya peningkatan energi tersebut diperlukan untuk perkembangan dan pertumbuhan janin. Gizi yang optimal sangat diperlukan selama awal kehamilan hingga menjelang persalinan, sehingga kehamilan aman dan dapat melahirkan dengan sehat. Janin yang ada dalam kandungan dan penyakit pada sang ibu sangat membutuhkan perhatian yang lebih. Apabila selama dalam kandungan terjadi kekurangan gizi maupun asupan gizi yang salah dapat mempengaruhi pertumbuhan dan perkembangan janin (Indriati & Sukaca, 2015).

2.3 Ibu hamil penderita Hipertensi

Hipertensi merupakan keadaan naiknya tekanan darah yang melebihi batas normal. Batas normal tekanan darah manusia adalah $\geq 140/90$ mmHg (Putri, 2018). Seorang wanita mengalami Hipertensi kehamilan jika tekanan darahnya melebihi batas normal tekanan darah tersebut (Putri, 2018).

Kondisi Hipertensi pada ibu hamil secara perlahan akan memberikan dampak yang tidak baik untuk ibu hamil dan janin yang ada dalam kandungannya. Dampak negatif tersebut antara lain:

- Berkurangnya aliran darah ke plasenta
- Perkembangan dan pertumbuhan janin terhambat, hal ini disebabkan janin kurang cukup menerima oksigen dan nutrisi untuk proses pertumbuhan.
- Dapat menyebabkan bayi lahir dengan berat badan yang rendah dan prematur.
- Dapat menyebabkan penyakit *kardiovaskular* apabila kondisi Hipertensi yang dibiarkan mencapai tahap *praeklamsia*.

2.4 Kebutuhan gizi ibu hamil

2.4.1 Kebutuhan energi

Kalori merupakan satuan yang digunakan untuk mengukur energi yang diperoleh tubuh dari asupan makanan yang dikonsumsi. Kandungan kalori dalam makanan dapat ditentukan oleh kandungan gizi yang ada dalam makanan

tersebut, buah-buahan misalnya yang memiliki kalori lebih rendah dibandingkan dengan nasi yang mengandung karbohidrat dan ayam yang mengandung lemak.

Asupan kalori yang dibutuhkan setiap orang berbeda-beda tergantung pada beberapa faktor yaitu tinggi badan, berat badan, jenis kelamin, usia dan aktivitas harian yang dilakukan, semakin banyak aktivitas yang dilakukan semakin banyak pula kalori yang dibutuhkan (Ariani, 2017). Oleh karena itu kebutuhan kalori harus diperhatikan oleh setiap orang agar tidak terjadi kekurangan maupun kelebihan kalori yang dapat mendatangkan penyakit. Kebutuhan energi ibu hamil berbeda di tiap usia kehamilannya yang dibagi menjadi tiga trimester kehamilan, sebelum menghitung kebutuhan energi ibu hamil terlebih dahulu menghitung berat badan ideal untuk menentukan status gizi menggunakan Persamaan (2.1).

$$BBI = 90\% \times (TB - 100) \times 1 \text{ kg} \quad (2.1)$$

Keterangan:

BBI : Berat badan ideal

TB : Tinggi badan (cm)

Setelah menghitung berat badan ideal selanjutnya menghitung angka metabolisme basal menggunakan Persamaan (2.2).

$$AMB = BBI \times 25 \text{ kalori} \quad (2.2)$$

Keterangan:

AMB : Angka metabolisme basal

BBI : Berat badan ideal

Berdasarkan hasil perhitungan BBI dapat ditentukan nilai berat yang akan digunakan pada perhitungan kalori berat. Nilai berat berdasarkan klasifikasi Berat Badan Ideal (BBI) ditunjukkan pada Tabel 2.2.

Tabel 2.2 Klasifikasi Berat Badan Ideal (BBI)

Nilai Berat Badan Ideal	Kategori Gizi	Nilai Berat (NB)
$BB < 90\%BBI$	Kurus	20%
$BB 90 - 110\% BBI$	Normal	0%
$BB 110-120\%$	Overweight	-10%
$BB > 120\% BBI$	Obesitas	-20%

Berdasarkan klasifikasi Berat Badan Ideal (BBI) maka dapat dihitung kalori berat menggunakan Persamaan (2.3).

$$\text{Kalori Berat} = NB \times AMB \quad (2.3)$$

Langkah selanjutnya adalah menghitung kalori harian yang dibutuhkan berdasarkan beberapa faktor antara lain:

- Usia

Usia mempengaruhi kebutuhan kalori harian yang dibutuhkan, nilai usia ditunjukkan pada Tabel 2.3.

Tabel 2.3 Nilai usia

Usia	Nilai Usia (NU)
40-59 tahun	-5%
60-69 tahun	-10%
≥70 tahun	-20%

Berdasarkan nilai usia pada Tabel 2.3 dapat dihitung kalori usia menggunakan Persamaan (2.4).

$$Kalori\ Usia = NU \times AMB \quad (2.4)$$

- Aktivitas fisik

Nilai aktivitas fisik berbeda-beda berdasarkan jenis pekerjaan yang dilakukan. Nilai aktivitas fisik ditunjukkan pada Tabel 2.4.

Tabel 2.4 Nilai aktivitas fisik

Nilai Aktivitas (NA)	Kategori	Pekerjaan
0.1	Istirahat	-
0.2	Ringan	Pegawai kantor, pegawai toko, ahli hukum, ibu rumah tangga.
0.3	Sedang	Pegawai di industri ringan, mahasiswa, militer yang sedang tidak perang.
0.4	Berat	Petani, buruh, militer dalam keadaan latihan, penari, atlit.
0.5	Sangat Berat	Tukang becak, tukang gali, pandai besi.

Berdasarkan nilai aktivitas fisik dapat dihitung kalori aktivitas menggunakan Persamaan (2.5).

$$Kalori\ Aktivitas = NA \times AMB \quad (2.5)$$

- Usia Kehamilan

Usia kehamilan memberikan tambahan kalori harian yang berbeda-beda seperti ditunjukkan pada Tabel 2.5.

Tabel 2.5 Nilai kalori kehamilan

Usia Kehamilan	Nilai Kalori Kehamilan (NK)
0-12 minggu	180 kalori

Tabel 2.5 Nilai kalori kehamilan (lanjutan)

Usia Kehamilan	Nilai Kalori Kehamilan (NK)
13-28 minggu	300 kalori
28-40 minggu	300 kalori

Berdasarkan penjelasan faktor-faktor yang mempengaruhi kebutuhan kalori, maka dapat dilakukan perhitungan total kalori harian menggunakan Persamaan (2.6).

$$\text{Kalori} = \text{AMB} + \text{Kalori Usia} + \text{Kalori Aktivitas} + \text{Kalori Berat} + \text{NK} \quad (2.6)$$

2.4.2 Karbohidrat

Karbohidrat merupakan tambahan kalori utama yang dibutuhkan selama masa kehamilan (Ariani, 2017). Karbohidrat diperlukan sebagai sumber kalori utama untuk pertumbuhan dan perkembangan janin. Jenis karbohidrat yang dianjurkan untuk dikonsumsi adalah karbohidrat kompleks seperti roti, sereal, nasi dan pasta (Ariani, 2017). Kebutuhan karbohidrat pada ibu hamil sebesar 65% dari total kebutuhan energi harian. Kebutuhan karbohidrat tersebut ditambah dengan kebutuhan karbohidrat sesuai dengan usia kehamilan yaitu Trimester I ditambah 20 gram, Trimester II dan III ditambah sebanyak 40 gram.

2.4.3 Protein

Protein merupakan nutrisi yang digunakan untuk proses pertumbuhan dan perkembangan janin. Protein digunakan untuk membentuk jaringan baru, maupun plasenta dan janin (Ariani, 2017). Protein juga berperan dalam pertumbuhan dan differensial sel (Ariani, 2017). Sumber protein didapat melalui protein hewani dan protein nabati (Ariani, 2017). Pada ibu hamil membutuhkan protein sebesar 15% dari total kebutuhan energi dalam satu hari ditambah sebanyak 20 gram di setiap trimester kehamilan.

2.4.4 Lemak

Lemak pada masa kehamilan dibutuhkan sebagai sumber tenaga yang vital dalam pertumbuhan jaringan plasenta (Ariani, 2017). Tubuh wanita hamil akan menyimpan lemak yang akan mendukung persiapan menyusui bayi setelah persalinan (Indriati & Sukaca, 2015). Lemak dibutuhkan untuk membentuk energy serta perkembangan sistem syaraf janin (Ariani, 2017). Kebutuhan lemak pada ibu hamil 20% dari total kebutuhan energi harian, Kebutuhan lemak tersebut ditambah dengan kebutuhan lemak sesuai dengan usia kehamilan yaitu Trimester I ditambah 6 gram, Trimester II dan III ditambah sebanyak 10 gram. Pengaturan tambahan kebutuhan gizi ibu hamil dapat dilihat pada Tabel 2.6.

Tabel 2.6 Tambahan kebutuhan gizi ibu hamil

Usia Kehamilan	Karbohidrat (gr)	Protein (gr)	Lemak (gr)
Trimester I	+25	+20	+6

Tabel 2.6 Tambahan kebutuhan gizi ibu hamil (lanjutan)

Usia Kehamilan	Karbohidrat (gr)	Protein (gr)	Lemak (gr)
Trimester II	+40	+20	+10
Trimester III	+40	+20	+10

2.4.5 Natrium

Pada ibu hamil penderita Hipertensi pengaturan kadar natrium mengikuti diet rendah garam 3 yaitu dengan memberikan kebutuhan natrium antara 1000-1200 mg per hari. Natrium mempengaruhi naiknya tekanan darah pada manusia, sehingga perlu diperhatikan, namun pada ibu hamil tidak mengikuti diet ketat rendah garam seperti pada umumnya karena hal itu akan berdampak pada kebutuhan protein yang dibutuhkan oleh ibu hamil menjadi berkurang.

2.5 Algoritme genetika

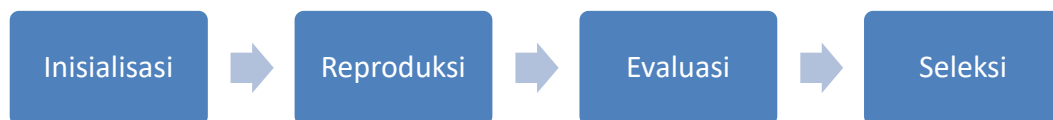
Algoritme genetika adalah algoritme yang menerapkan konsep evolusi alamiah yang dimulai dengan kumpulan solusi yang disebut populasi kemudian disusun menjadi individu. Algoritme ini dapat digunakan untuk menyelesaikan masalah optimasi dengan model matematika yang kompleks dan sulit dibangun (Mahmudy, 2015). Algoritme genetika (AG) memiliki karakteristik yang membedakan dengan prosedur optimasi yang lain yaitu sebagai berikut:

1. AG menggunakan pengkodean dari himpunan solusi berdasarkan parameter yang ditetapkan.
2. AG mencari solusi pada sebuah populasi dari sejumlah individu-individu.
3. AG menggunakan nilai *fitness* dalam mengevaluasi individu yang merupakan solusi terbaik.
4. AG menggunakan aturan-aturan transisi peluang, bukan aturan-aturan deterministik.

Pada algoritme genetika ada beberapa hal yang harus dilakukan antara lain (Diri, 2015):

1. Mendefinisikan individu, individu merupakan salah satu solusi yang mungkin dari permasalahan
2. Mendefinisikan nilai *fitness* yaitu ukuran baik tidaknya sebuah idividu atau solusi yang didapatkan
3. Menentukan bagaimana proses pembangkitan populasi awal
4. Menentukan proses seleksi yang akan dilakukan
5. Menentukan proses *crossover* (perkawinan silang) dan mutasi gen.

Tahapan proses pada algoritme genetika dapat dilihat pada Gambar 2.1.



Gambar 2.1 Alur Algoritme Genetika

2.5.1 Inisialisasi

Proses inisialisasi bertujuan membangkitkan solusi baru secara acak dari sejumlah kromosom dan diletakkan pada populasi. Terlebih dahulu menentukan ukuran populasi atau *popsiz*e yang menyatakan jumlah kromosom pada populasi (Mahmudy, 2013).

2.5.2 Reproduksi

Proses reproduksi bertujuan menghasilkan individu baru berdasarkan individu-individu yang sudah ada pada populasi, individu baru tersebut diletakkan pada *offspring* (Mahmudy, 2013).

A. Crossover

Crossover merupakan reproduksi pertama yang dilakukan pada algoritme genetika. Proses *crossover* melibatkan dua induk dengan harapan keturunan yang dihasilkan lebih baik daripada induknya. Pada penelitian ini akan menggunakan *extended intermediate crossover*. Proses ini diawali dengan memilih acak dua induk pada populasi yang akan menghasilkan *offspring*. Banyaknya proses *crossover* yang dilakukan berdasarkan hasil kali *crossover rate* dengan *popsiz*e. Satu kali proses *crossover* akan menghasilkan 2 *offspring*. Nilai setiap gen pada kedua *offspring* dapat ditentukan menggunakan Persamaan (2.7) dan Persamaan (2.8) (Mahmudy, 2015).

$$C_1 = P_1 + \alpha (P_2 - P_1) \quad (2.7)$$

$$C_2 = P_2 + \alpha (P_1 - P_2) \quad (2.8)$$

Keterangan:

P_1 : induk *crossover* ke-1

P_2 : induk *crossover* ke-2

C_1 : *offspring* ke-1

C_2 : *offspring* ke-2

Contoh proses *extended intermediate crossover* ditunjukkan oleh Gambar 2.2.

	X_1	X_2	X_3
P_1	0.2105	0.4043	0.1037
P_2	0.3015	0.1304	0.2526
α	0.1047	0.2372	0.2101

↓

C_1	0.2200	0.3393	0.1350
C_2	0.2920	0.1954	0.2213

Gambar 2.2 Contoh *extended intermediate crossover*

B. Mutasi

Mutasi dilakukan setelah proses *crossover*, yang mana proses ini memilih acak satu kromosom yang akan dimutasi. Jumlah *offspring* yang dihasilkan ditentukan berdasarkan hasil kali *mutation rate* dengan *popsize*. Satu kali proses mutasi akan menghasilkan satu *offspring*. Pada penelitian ini akan menggunakan metode *Random Mutation*. *Random mutation* diawali dengan memilih acak satu buah individu, kemudian menentukan gen yang akan dimutasi serta nilai maksimum dan minimum dari kromosom induk terpilih. Pada random mutation dibangkitkan nilai r dari interval bilangan tertentu. Nilai gen *offspring* dapat ditentukan menggunakan Persamaan (2.9) (Mahmudy, 2015).

$$X'_i = X_i + r (Max_i - Min_i) \quad (2.9)$$

Contoh proses menggunakan *random mutation* ditentukan gen yang dimutasi yaitu X_3 dengan nilai $r = -0.0357$, $Max_i = 0.5036$ dan $Min_i = 0.0172$, maka proses *random mutation* dan *offspring* yang dihasilkan dapat dilihat pada Gambar 2.3.

$$X_1 = 0.2105 \text{ (tetap)}$$

$$X_2 = 0.4043 \text{ (tetap)}$$

$$X_3 = 0.1037 + (-0.0357 \times (0.5036 - 0.0172)) = 0.0863$$

	X_1	X_2	X_3
P_1	0.2105	0.4043	0.1037

↓

C_1	0.2105	0.4043	0.0863
-------	--------	--------	--------

Gambar 2.3 Contoh *random mutation*

2.5.3 Evaluasi

Evaluasi merupakan proses untuk mengetahui kualitas individu-individu dalam popuasi (Mahmudy, 2013). Nilai *fitness* merupakan nilai baik tidaknya suatu individu, dan dijadikan acuan dalam pencapaian nilai optimal yaitu nilai *fitness* yang paling tinggi (Sari, 2014). Sebelum menghitung nilai *fitness* terlebih dahulu

menghitung penalti yang merupakan kondisi terdapat nilai yang tidak sama dengan aturan. Untuk menghitung nilai *fitness* pada kasus maksimasi dapat menggunakan Persamaan (2.10) (Mahmudy, 2015).

$$f = h \quad (2.10)$$

Keterangan:

f = nilai *fitness*

h = rumus fungsi maksimasi yang digunakan

Untuk kasus minimasi, dapat menggunakan persamaan (2.11):

$$f = \frac{1}{h} \quad (2.11)$$

Perhitungan nilai *fitness* yang digunakan pada optimasi asupan makanan harian ibu hamil penderita *Hipertensi* ditunjukkan pada Persamaan (2.12).

$$Fitness = \frac{C1}{1+\sum PK} + \frac{C2}{1+\sum PKb} + \frac{C3}{1+\sum PP} + \frac{C4}{1+\sum PL} + \frac{C5}{1+\sum PN} + \frac{C6}{\sum harga} + C.V \quad (2.12)$$

Nilai penalti yang digunakan pada perhitungan nilai *fitness* berdasarkan penalti masing-masing kebutuhan gizi pada Persamaan (2.13) sampai (2.17).

$$PK = \begin{cases} NKS - NKA, & NKS \geq NKA \\ NKA - NKS, & NKS < NKA \end{cases} \quad (2.13)$$

$$PKb = \begin{cases} NKbS - NKbA, & NKbS \geq NKbA \\ NKbA - NKbS, & NKbS < NKbA \end{cases} \quad (2.14)$$

$$PP = \begin{cases} NPS - NPA, & NPS \geq NPA \\ NPA - NPS, & NPS < NPA \end{cases} \quad (2.15)$$

$$PL = \begin{cases} NLS - NLA, & NLS \geq NLA \\ NLA - NLS, & NLS < NLA \end{cases} \quad (2.16)$$

$$PN = \begin{cases} 0, & 1000 \leq NNS \leq 1200 \\ 1000 - NPS, & NNS \leq 1000 \\ NPS - 1200, & NNS \geq 1200 \end{cases} \quad (2.17)$$

Keterangan:

PK : Penalti kalori

PKb : Penalti karbohidrat

PP : Penalti protein

PL : Penalti lemak

PN : Penalti natrium

NKS : Nilai kalori sistem

NKbS : Nilai karbohidrat sistem

NPS : Nilai protein sistem

NLS : Nilai lemak sistem

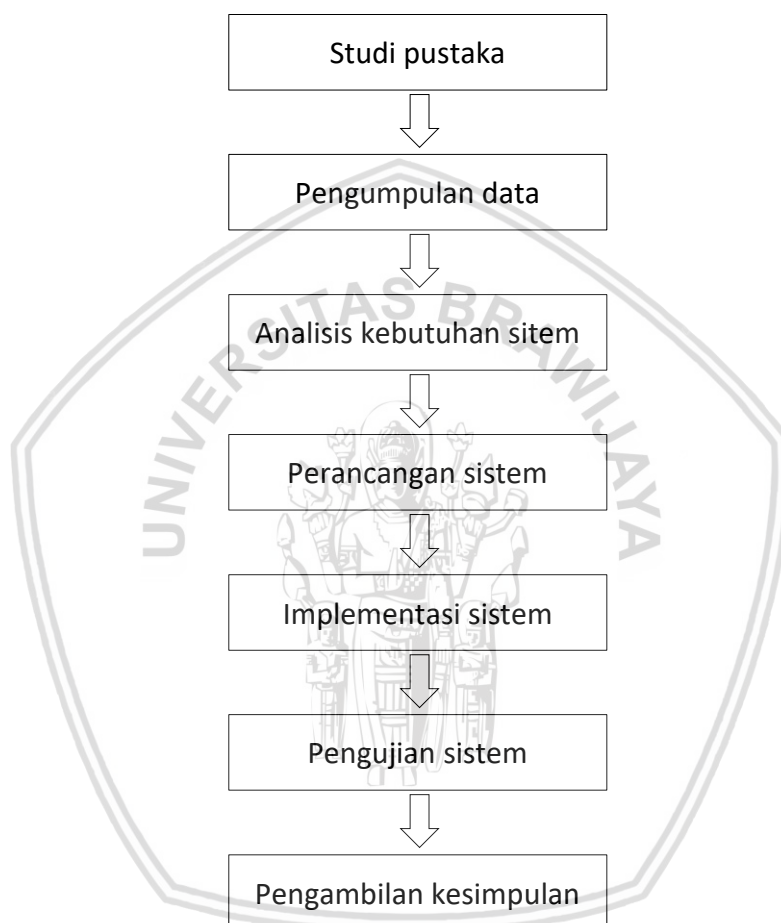
NNS	: Nilai natrium sistem
NKA	: Jumlah kebutuhan kalori aktuan
NKba	: Jumlah kebutuhan karbohidrat aktual
NPA	: Jumlah kebutuhan protein aktual
NLA	: Jumlah kebutuhan lemak aktual
NNA	: Nilai kebutuhan natrium actual
C1	: Konstanta kalori
C2	: Konstanta karbohidrat
C3	: Konstanta protein
C4	: Konstanta lemak
C5	: Konstanta natrium
C6	: Konstanta harga
C	: Konstanta variasi

2.5.4 Seleksi

Setiap kromosom pada populasi akan melalui proses seleksi. Tujuan seleksi adalah untuk mendapatkan calon generasi yang baik. Langkah awal dalam seleksi yaitu mengurutkan individu berdasarkan nilai *fitness* terbesar hingga terkecil.. Nilai *fitness* yang besar dari sebuah kromosom membuat semakin besar peluang terpilihnya kromosom tersebut untuk menghasilkan generasi yang lebih baik lagi selanjutnya. Metode seleksi yang digunakan dalam penelitian ini adalah *elitism*. *Elitism* merupakan metode seleksi yang mengumpulkan individu-individu dalam populasi dan *offspring* pada satu penampungan, nilai *fitness* yang paling baik dalam penampungan akan masuk pada generasi berikutnya (Mahmudy, 2015).

BAB 3 METODOLOGI

Pada bab ini akan dijelaskan mengenai tahapan-tahapan yang dilakukan selama penelitian. Tahapan dalam penelitian ini dilakukan secara sistematis, meliputi studi pustaka, pengumpulan data, analisis kebutuhan sistem, perancangan sistem, implementasi sistem, pengujian sistem dan pengambilan kesimpulan. Tahapan penelitian yang akan dilakukan ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram alir penelitian

3.1 Studi pustaka

Studi pustaka dilakukan sebagai dasar pemahaman terhadap teori-teori yang berkaitan dengan permasalahan dalam penelitian. Teori-teori tersebut diperoleh dari berbagai sumber yaitu pakar, penelitian yang telah dilakukan sebelumnya, dan dokumentasi internet. Teori pendukung yang berkaitan dengan penelitian ini antara lain:

a. Kehamilan

Teori pendukung tentang kehamilan yang digunakan berisi penjelasan mengenai pengelompokan usia kehamilan, kebutuhan gizi ibu hamil dan kebutuhan energi ibu hamil.

- b. Ibu hamil penderita Hipertensi
Berisi penjelasan mengenai Hipertensi serta dampak yang menyerang janin dan ibu hamil.
- c. Algoritme Genetika
Dasar teori algoritme genetika yang digunakan berisi penjelasan mengenai algoritme genetika dan tahapan-tahapan dalam penyelesaian masalah.

3.2 Pengumpulan data

Pengumpulan data dalam penelitian ini dilakukan untuk memperoleh informasi mengenai data-data yang diperlukan dalam penelitian. Data yang digunakan dalam penelitian ini adalah data ibu hamil penderita Hipertensi yang didapatkan dari Bidan Praktek Swasta Mojokerto dan data makanan beserta kandungan gizinya. Dalam pengumpulan data dilakukan wawancara dengan pakar yaitu seorang ahli gizi dari RS Saiful Anwar Malang bernama Nawangsari L. untuk memperoleh informasi yang berkaitan dengan kehamilan meliputi pengelompokan usia kehamilan dan menu makanan yang dibutuhkan bagi ibu hamil penderita Hipertensi. Data makanan yang didapatkan dalam wawancara tersebut adalah data makanan olahan dengan kandungan gizi bersumber dari program Nutrisurvey.

3.3 Analisis kebutuhan sistem

Analisis kebutuhan sistem dilakukan untuk mengetahui kebutuhan apa saja yang diperlukan dalam membangun sistem meliputi kebutuhan hardware, kebutuhan software dan kebutuhan data. Kebutuhan hardware merupakan kebutuhan perangkat keras yang digunakan dalam membangun sistem yaitu perangkat komputer atau laptop beserta spesifikasinya. Kebutuhan software yaitu perangkat lunak meliputi sistem operasi yang mendukung, dan bahasa pemrograman. Kebutuhan data meliputi data makanan dan data ibu hamil. Setiap kebutuhan tersebut harus dipenuhi guna melakukan perancangan, implementasi, dan pengujian sistem. Perangkat yang digunakan dalam penelitian ini antara lain:

- a. Perangkat keras:
 - Laptop dengan processor Intel® Core™ i7-3610QM CPU @2.30GHz 2.30GHz
 - RAM 6 GB
- b. Perangkat Lunak:
 - Sistem Operasi Windows 10
 - Netbeans IDE 8.0.2

3.4 Perancangan sistem

Pada perancangan sistem ini dilakukan berdasarkan seluruh kebutuhan sistem yang sudah dijelaskan. Dalam perancangan akan dijelaskan bagaimana *input* akan diproses menjadi output yang tepat. Sistem yang dibuat yaitu optimasi asupan makanan harian ibu hamil penderita Hipertensi akan memproses variabel *input*

yang dimasukkan dengan menggunakan algoritme genetika yang kemudian akan menghasilkan output sebuah kesimpulan asupan makanan dalam beberapa hari.

3.5 Implementasi sistem

Implementasi sistem ini akan dilakukan dengan mengacu pada hasil analisis kebutuhan sistem dan perancangan sistem. Sistem akan dibangun menggunakan bahasa pemrograman Java, dan menerapkan algoritme genetika.

3.6 Pengujian sistem

Pengujian sistem dilakukan untuk menentukan parameter algoritme genetika yang tepat untuk digunakan dalam optimasi asupan makanan harian ibu hamil penderita Hipertensi yaitu dengan melakukan pengujian terhadap ukuran generasi, ukuran populasi, dan kombinasi nilai *cr* dan *mr*. Selain itu pengujian juga dilakukan untuk mengetahui kualitas hasil rekomendasi makanan yang dihasilkan oleh sistem.

3.7 Pengambilan kesimpulan

Tahap terakhir yang dilakukan dalam penelitian ini yaitu pengambilan kesimpulan. Pengambilan kesimpulan dalam sistem ini dilakukan dengan memperhatikan setiap tahapan meliputi perancangan, implementasi, dan pengujian yang telah dilakukan. Kesimpulan optimasi asupan makanan harian ibu hamil penderita Hipertensi ini berdasarkan hasil analisa pengujian sistem. Kesimpulan yang dihasilkan akan menjawab rumusan masalah pada penelitian ini.

BAB 4 PERANCANGAN

4.1 Formulasi permasalahan

Permasalahan yang akan diselesaikan adalah optimasi asupan makanan harian ibu hamil penderita *Hipertensi* dengan menggunakan data makanan beserta kandungan gizinya. Data makanan yang digunakan terdiri dari 5 kategori yaitu sumber karbohidrat berjumlah 22 jenis, sumber protein hewani berjumlah 25 jenis, sumber protein nabati berjumlah 17 jenis, sayuran berjumlah 23 jenis, dan buah-buahan berjumlah 23 jenis. Setiap kategori makanan memiliki anjuran porsi yang berbeda-beda, untuk sumber karbohidrat anjuran porsi sebanyak 180 gram, protein hewani sebanyak 50 gram, protein nabati 50 gram dan 100 gram, sayuran sebanyak 200 gram, dan buah-buahan sebanyak 200 gram. Contoh data makanan sumber karbohidrat yang digunakan beserta kandungan gizinya dapat dilihat pada Tabel 4.1 untuk data lengkap yang digunakan dapat dilihat pada Lampiran A.

Tabel 4.1 Sumber karbohidrat

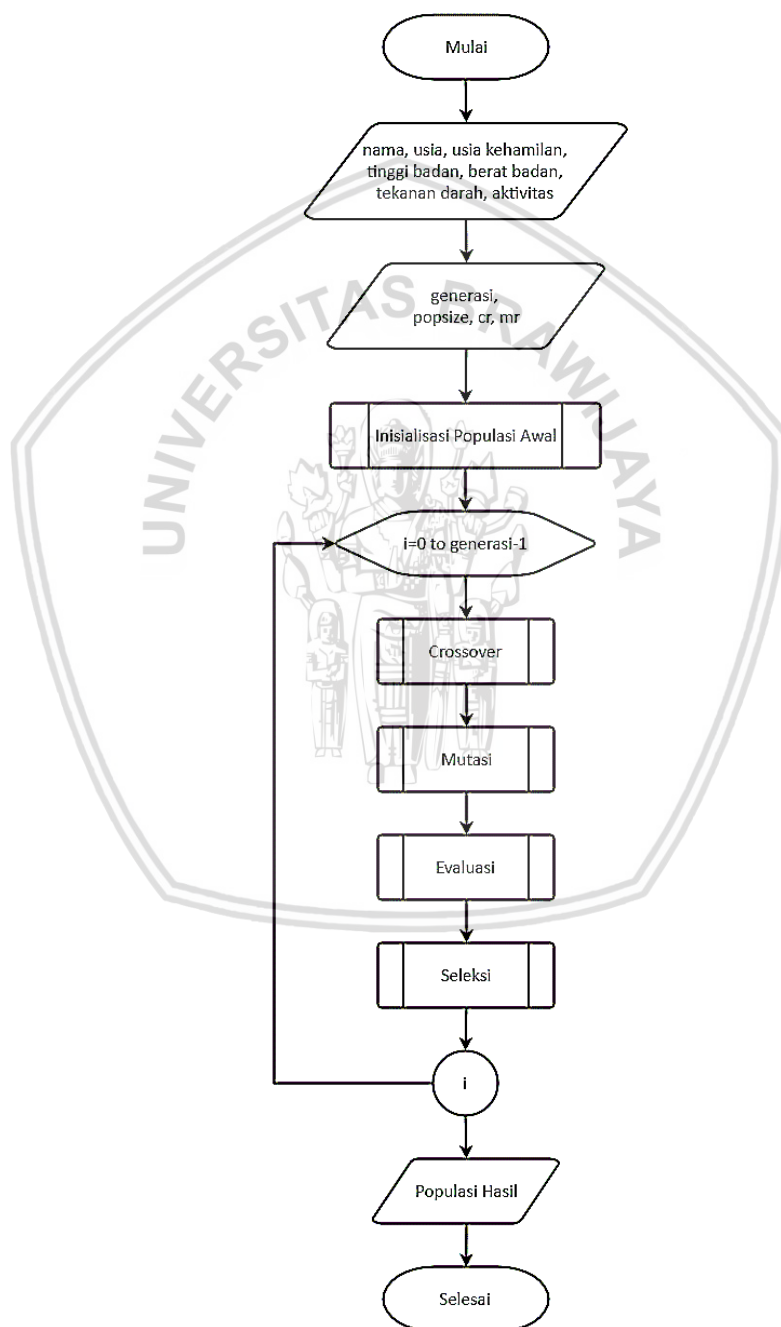
No.	Nama Makanan	Berat (gr)	Kalori (kkal)	Karbohidrat (gr)	Protein (gr)	Lemak (gr)	Natrium (mg)
1	Nasi Putih	180	234.0	51.5	4.3	0.4	0.0
2	Nasi Goreng	180	450.0	36.2	6.3	31.0	16.2
3	Nasi Jagung	180	217.7	49.0	5.0	1.1	12.6
4	Nasi Tim	180	210.8	46.3	4.0	0.4	0.0
5	Nasi Uduk	180	212.5	42.3	3.8	2.7	1.8
6	Bubur Nasi	180	131.2	28.8	2.3	0.2	0.0
7	Bubur Tim	180	142.4	29.0	3.6	0.9	10.8
8	Bubur Havermoth	180	111.4	19.4	4.7	1.8	1.8

4.2 Siklus penyelesaian masalah

Pada proses penyelesaian masalah ini terdapat beberapa langkah yang dilakukan. Langkah pertama yaitu inisialisasi parameter, parameter pada permasalahan ini terdiri dari parameter data ibu hamil dan parameter algoritme genetika. Selanjutnya parameter data ibu hamil akan diproses oleh sistem untuk mendapatkan hasil perhitungan kebutuhan energi harian yang harus dipenuhi. Parameter yang kedua yaitu parameter algoritme genetika akan digunakan oleh sistem dalam melakukan optimasi asupan makanan harian yang akan dikonsumsi berdasarkan hasil perhitungan kebutuhan energi. Hasil akhir yang akan ditampilkan oleh sistem adalah rekomendasi makanan yang dapat dikonsumsi oleh ibu hamil dalam beberapa hari.

4.3 Tahapan algoritme genetika

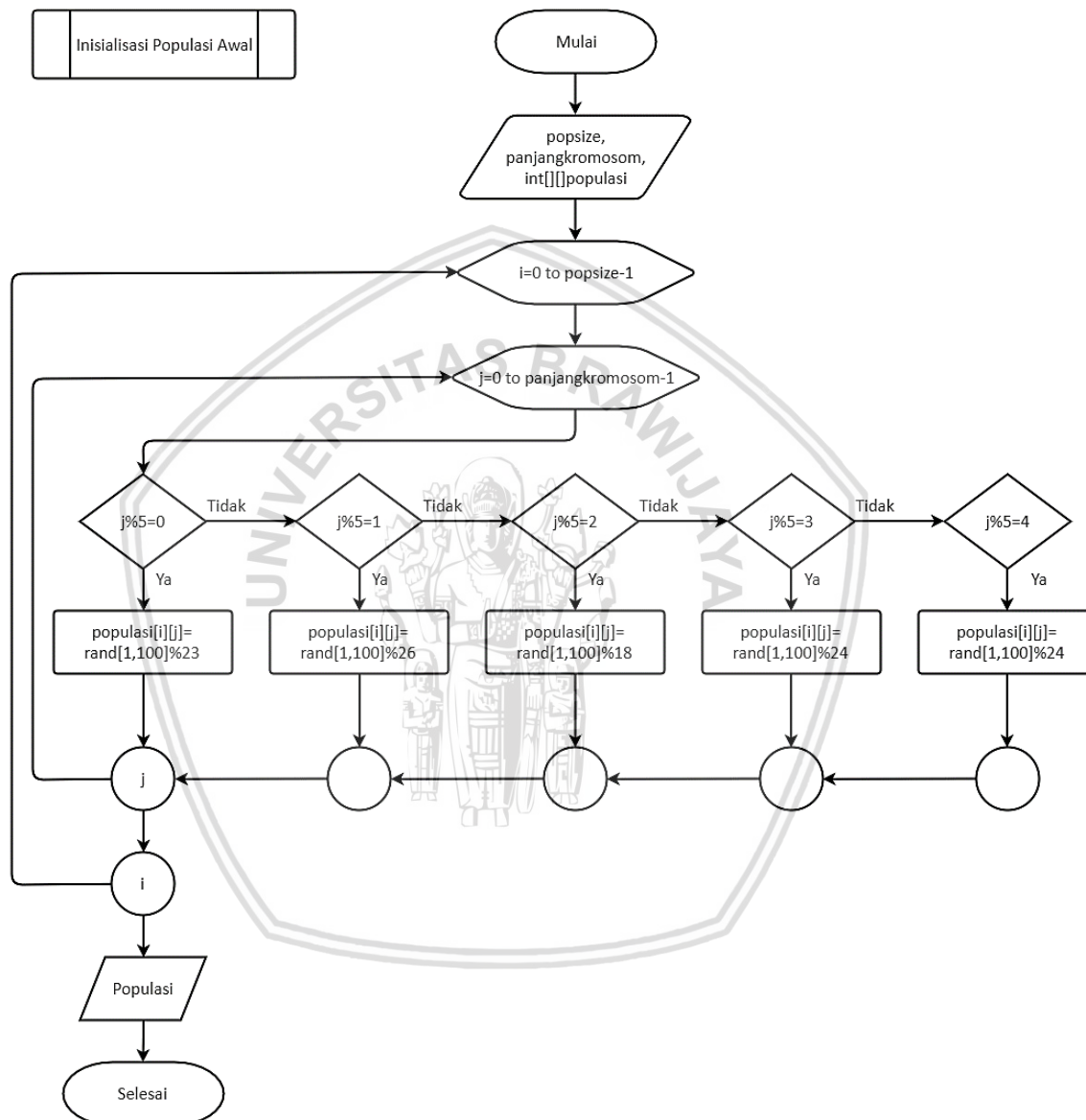
Pada algoritme genetika terdapat beberapa tahapan yang dilakukan yaitu inialisasi populasi awal untuk mendapatkan individu-individu sebagai calon solusi, selanjutnya tahapan reproduksi yang terdiri dari *crossover* dan mutasi untuk menghasilkan keturunan baru, kemudian tahap evaluasi untuk menghitung nilai *fitness*, dan yang terakhir adalah seleksi untuk mendapatkan individu-individu pada generasi selanjutnya sekaligus individu terbaik. Berikut diagram alir algoritme genetika ditunjukkan pada Gambar 4.1.



Gambar 4.1 Diagram alir algoritme genetika

4.3.1 Inisialisasi populasi awal

Tahap awal yang dilakukan adalah Inisialisasi populasi awal untuk mendapatkan individu-individu calon solusi sebanyak *popsi*. Diawali dengan menentukan nilai *popsi*, kemudian membangkitkan bilangan integer secara acak sebagai nilai gen yang merepresentasikan id makanan. Berikut diagram alir proses inisialisasi populasi awal pada permasalahan ini ditunjukkan pada Gambar 4.2.

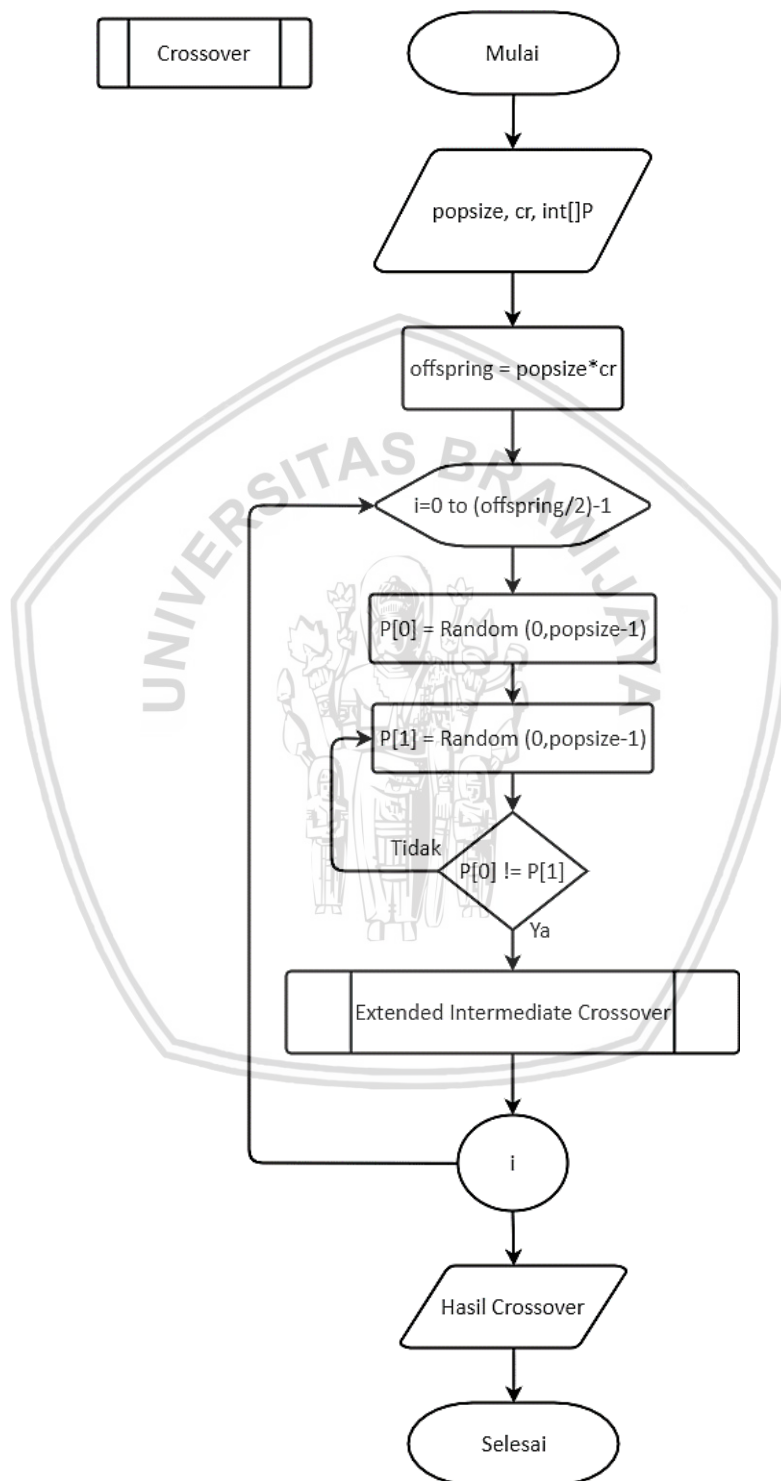


Gambar 4.2 Diagram alir inisialisasi populasi awal

Pada Gambar 4.2 proses inisialisasi populasi awal setiap gen dalam kromosom direpresentasikan dalam bilangan integer acak dengan batas bilangan maksimum sama dengan jumlah sumber makanan yang direpresentasikan ditambah satu. Terdapat perulangan *i* yang merupakan perulangan untuk membangkitkan individu sebanyak *popsi*, dan perulangan *j* merupakan perulangan untuk menentukan nilai setiap gen pada kromosom.

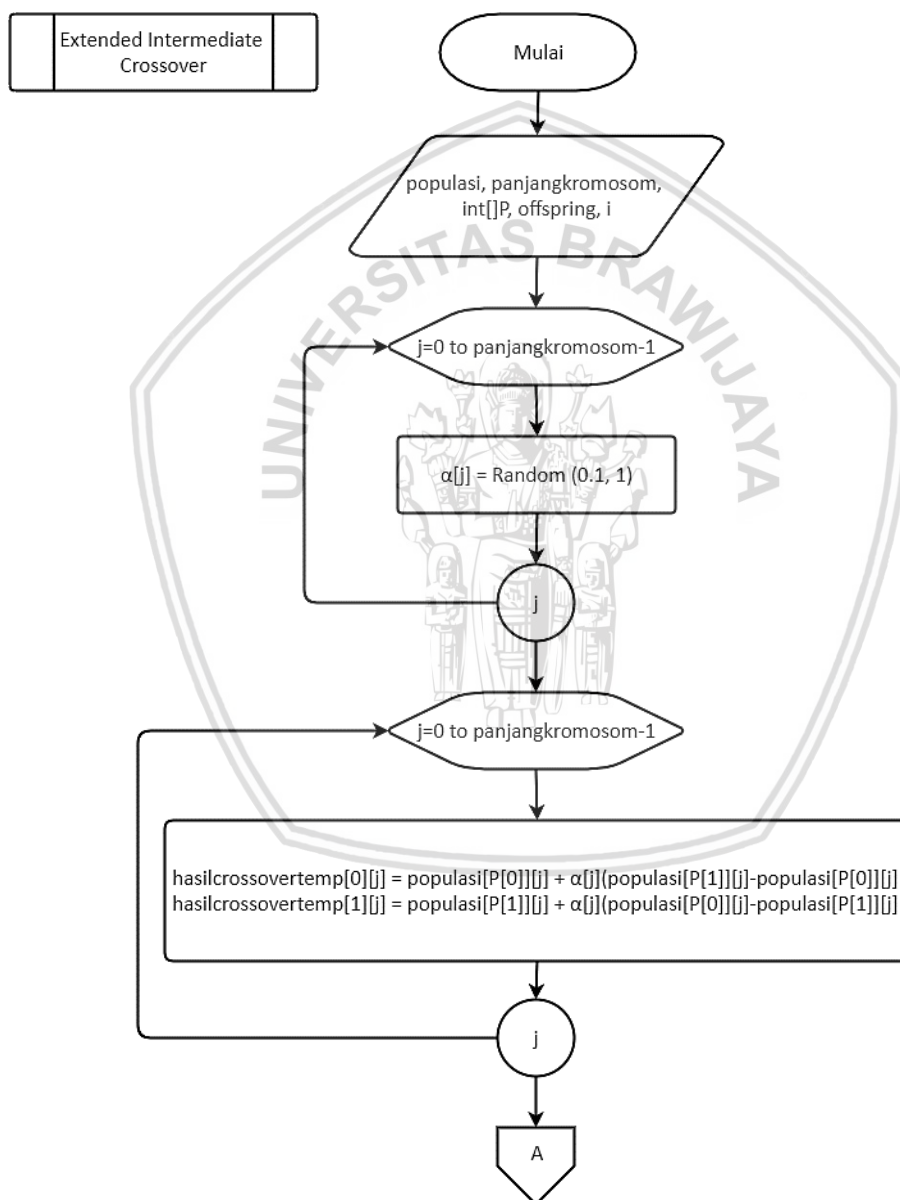
4.3.2 Crossover

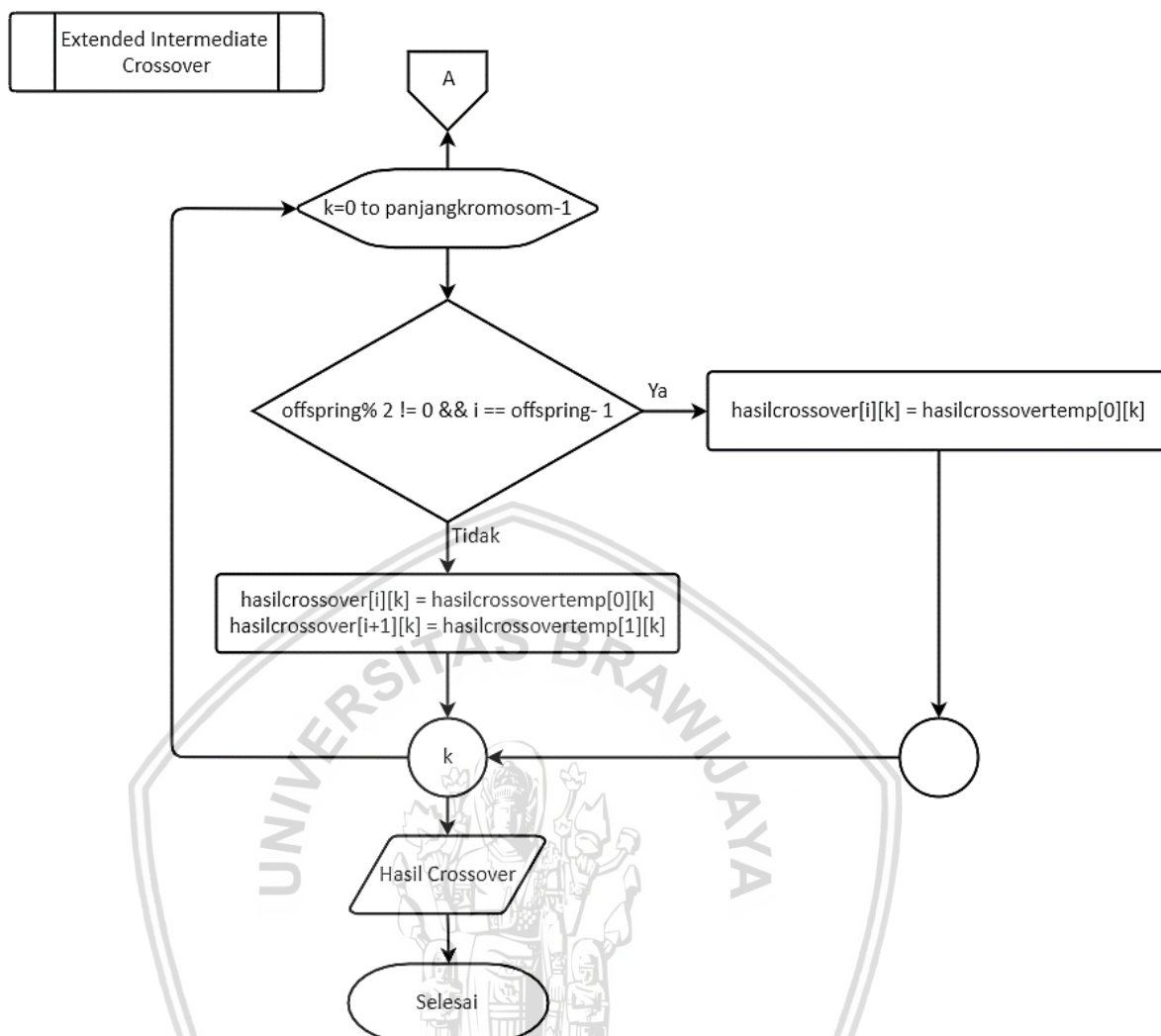
Proses *crossover* pada penyelesaian permasalahan ini menggunakan metode *extended intermediate crossover*. Berikut diagram alir proses *crossover* ditunjukkan pada Gambar 4.3.



Gambar 4.3 Diagram alir proses *crossover*

Gambar 4.3 menunjukkan proses *crossover*. Proses *crossover* diawali dengan menentukan nilai offspring yang akan dihasilkan yaitu hasil kali *crossover rate* (*cr*) dengan *popsiz*e. Terdapat perulangan *i* sebanyak jumlah *offspring/2* sebagai batas perulangan proses *extended intermediate crossover*. Pemilihan induk *crossover* dilakukan secara acak dengan memilih individu *P* sebanyak 2 buah. Dua individu *P* yang digunakan pada proses *crossover* harus berbeda. Apabila dihasilkan 2 individu yang sama maka individu kedua akan dipilih ulang secara acak hingga tidak sama dengan individu pertama. Setelah mendapatkan 2 individu yang berbeda selanjutnya akan masuk pada proses *extended intermediate crossover* seperti ditunjukkan pada Gambar 4.4.



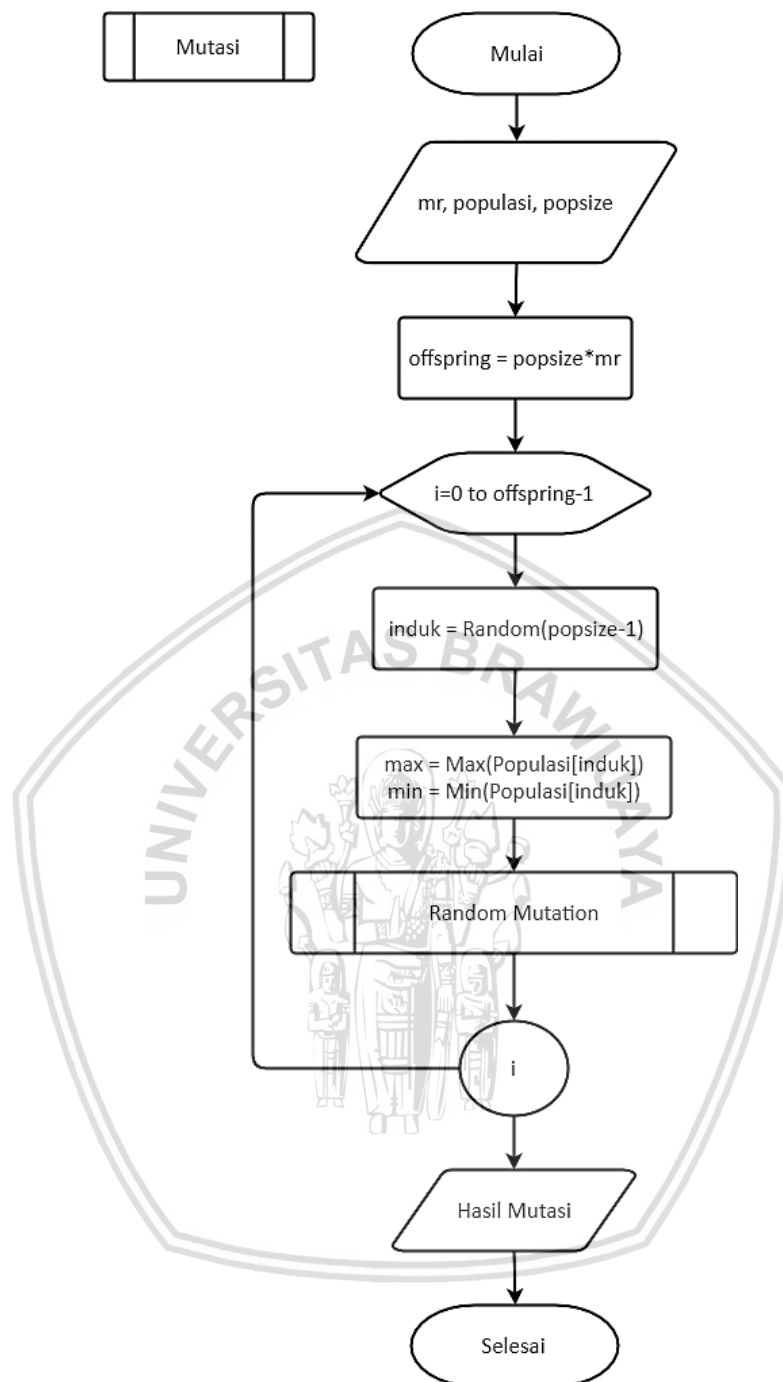


Gambar 4.4 Diagram alir extended intermediate crossover

Pada Gambar 4.4 proses *extended intermediate crossover* diawali dengan perulangan membangkitkan nilai α dalam interval bilangan 0.1 hingga 1 secara acak sebanyak panjang kromosom. Selanjutnya masuk pada perulangan untuk menentukan nilai gen dua buah *offspring* yang dihasilkan berdasarkan nilai α sebanyak panjang kromosom dan disimpan dalam *hasilcrossovertemp*. Kemudian diberikan kondisi apabila *offspring* yang dihasilkan bernilai ganjil maka *offspring* yang masuk pada penampungan *hasilcrossover* dalam proses *crossover* yang terakhir adalah *offspring* yang pertama terbentuk, sedangkan apabila jumlah *offspring* yang dihasilkan genap maka kedua *offspring* tersebut masuk ke dalam penampungan *hasilcrossover*.

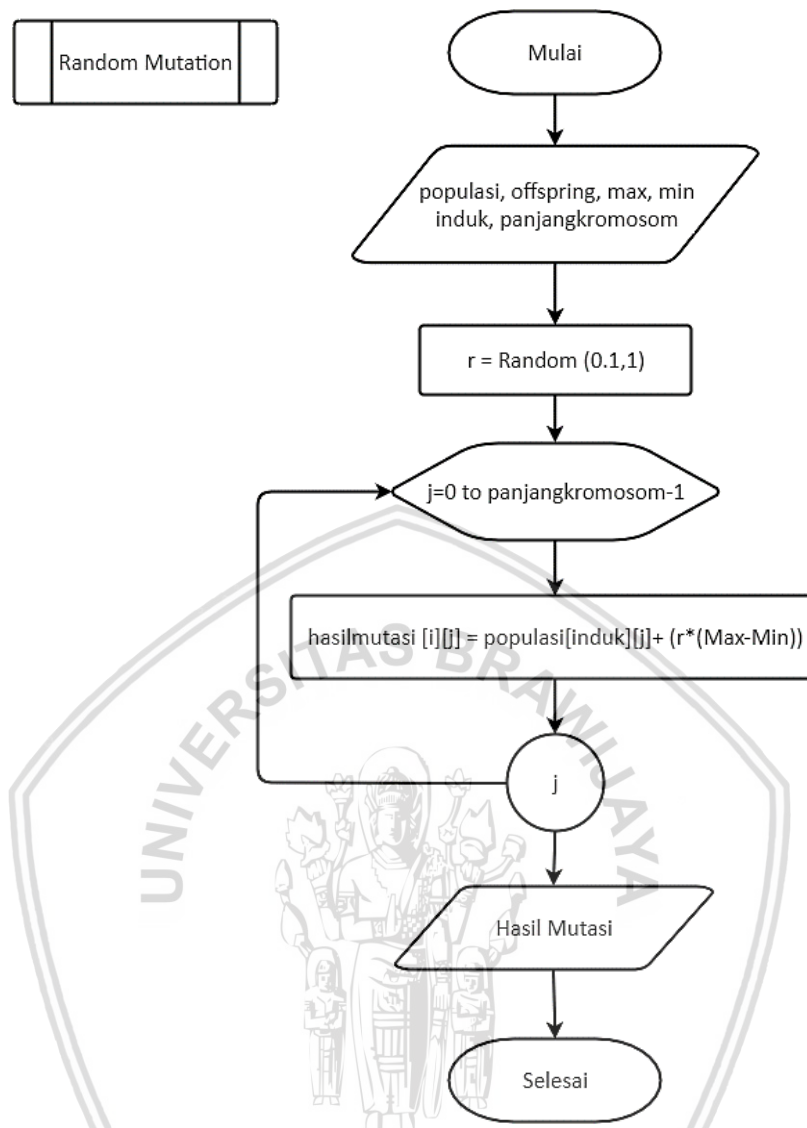
4.3.3 Mutasi

Metode mutasi yang digunakan adalah *random mutation*. Berikut diagram alir proses mutasi pada penelitian ini ditunjukkan oleh Gambar 4.5.



Gambar 4.5 Diagram alir proses mutasi

Pada Gambar 4.5 proses mutasi diawali dengan menentukan banyaknya *offspring* yang akan dihasilkan yaitu hasil kali *mutation rate* (*mr*) dengan *popsi*. Kemudian memilih satu individu secara acak untuk dimutasi. Selanjutnya menentukan nilai maksimum dan minimum induk yang terpilih untuk digunakan pada proses *random mutation*. Diagram alir proses *random mutation* pada penelitian ini ditunjukkan pada Gambar 4.6.

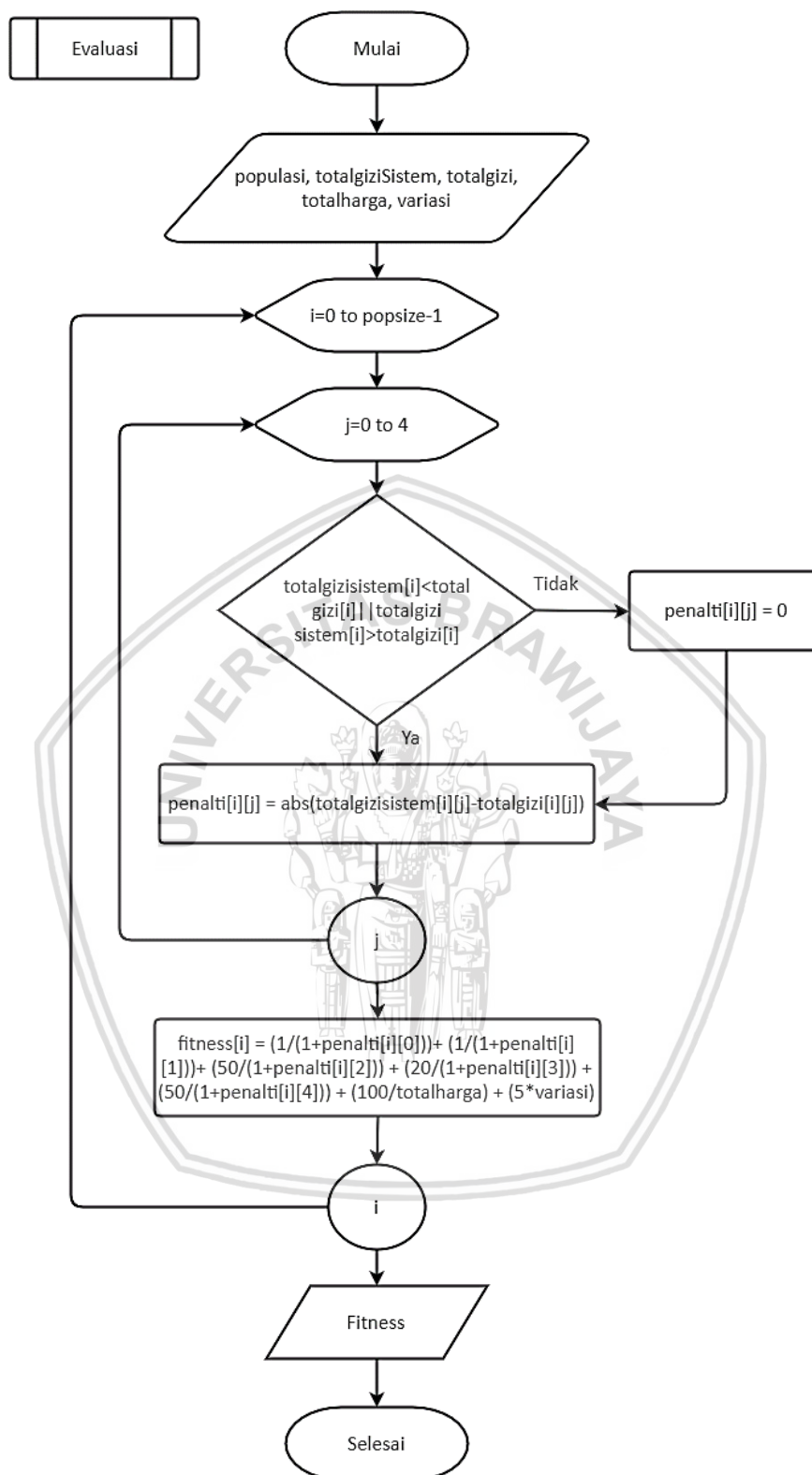


Gambar 4.6 Diagram alir *random mutation*

Pada Gambar 4.6 proses *random mutation* diawali dengan membangkitkan nilai r secara acak dengan interval antara 0.1 hingga 1. Kemudian masuk pada perulangan j sebanyak panjang kromosom untuk menentukan nilai gen *offspring* mutasi yaitu hasil penjumlahan gen ke- j dengan hasil kali bilangan r dengan nilai maksimum dikurangi minimum. Gen yang akan dimutasi adalah semua gen yang ada pada kromosom induk terpilih.

4.3.4 Evaluasi

Proses evaluasi dilakukan untuk menghitung nilai *fitness* masing-masing individu. Diawali dengan menghitung nilai penalti masing-masing gizi, total harga dan total variasi. Diagram alir proses evaluasi ditunjukkan oleh Gambar 4.7.



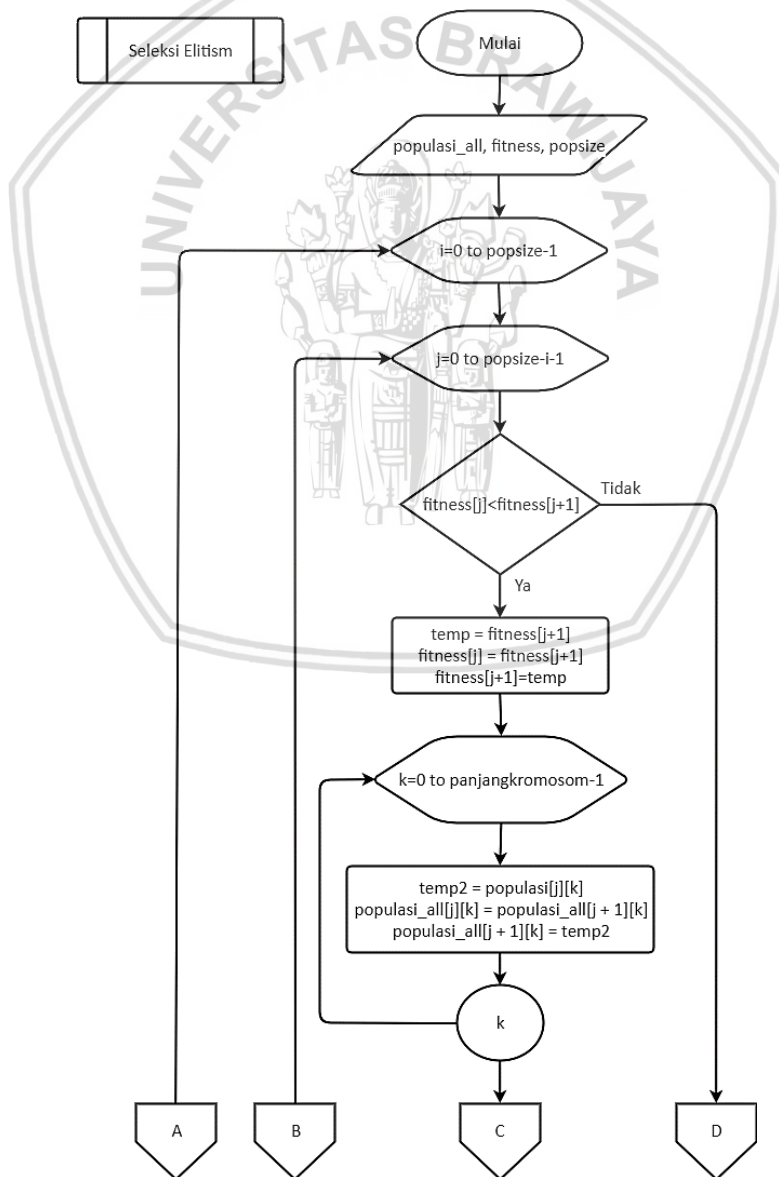
Gambar 4.7 Diagram alir evaluasi

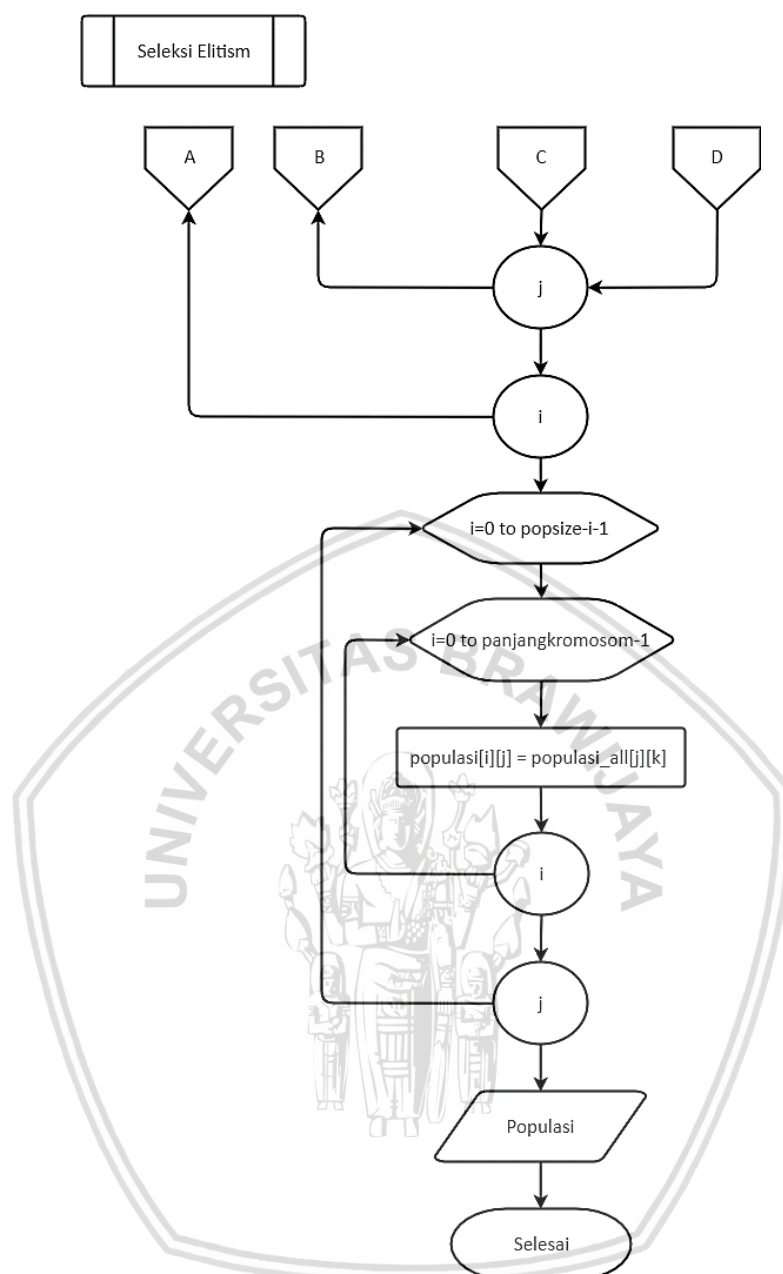
Pada Gambar 4.7 proses evaluasi terdapat perulangan *i* sebanyak *popsi* untuk menghitung nilai *fitness* masing-masing individu, kemudian perulangan *j* untuk menghitung nilai penalti kandungan gizi pada masing-masing gen. Apabila

nilai gen ke-j yaitu kandungan gizi rekomendasi sistem kurang dari kebutuhan gizi aktual atau sebaliknya maka nilai penalti sama dengan absolut kandungan gizi sistem dikurangi kebutuhan gizi aktual, sedangkan jika tidak memenuhi kondisi tersebut maka nilai penalti sama dengan nol. Kemudian menghitung nilai fitness yaitu hasil penjumlahan setiap konstanta dibagi penalti gizi, ditambah dengan konstanta dibagi jumlah harga, dan ditambah dengan konstanta dikalikan dengan variasi.

4.3.5 Seleksi

Proses seleksi yang dilakukan pada penyelesaian permasalahan ini menggunakan metode *elitism*. Awalnya populasi awal dan *offspring* hasil reproduksi dikumpulkan dalam satu penampungan, lalu mengurutkan berdasarkan nilai *fitness* tertinggi, dan mengambil individu dengan *fitness* terbaik sebanyak *popsi*. Individu dengan nilai *fitness* tertinggi terpilih sebagai solusi. Berikut diagram alir proses seleksi *elitism* ditunjukkan oleh Gambar 4.8.





Gambar 4.8 Diagram alir seleksi *elitism*

Pada Gambar 4.8 proses seleksi diawali dengan mengurutkan nilai *fitness* setiap individu dari yang terbesar hingga terkecil. Kemudian dipilih individu dengan nilai *fitness* tertinggi sebanyak *popsi* sebagai populasi baru dan individu dengan nilai *fitness* paling tinggi dipilih sebagai solusi.

4.4 Contoh perhitungan manual

4.4.1 Contoh permasalahan

Diberikan contoh permasalahan ibu hamil penderita Hipertensi dengan parameter data *input* sebagai berikut:

- Nama : Y
- Usia : 28 tahun
- Usia kehamilan : 8 minggu
- Tinggi badan : 156 cm
- Berat badan : 50 kg
- Tekanan darah : 160/100 mmHg
- Aktivitas : Ringan

Berdasarkan parameter *input* di atas maka dapat dihitung kebutuhan energi pada ibu hamil tersebut dengan langkah pertama yang harus dilakukan adalah menghitung Berat Badan Ideal (BBI) menggunakan Persamaan (2.1).

$$BBI = 90\% \times (156 - 100) \times 1 = 50.4 \text{ kg}$$

Berdasarkan hasil perhitungan BBI maka ibu hamil tersebut tergolong dalam kategori normal. Selanjutnya menghitung Angka Metabolisme Basal (AMB) menggunakan Persamaan (2.2).

$$AMB = 50.4 \times 25 = 1260 \text{ kkal}$$

Setelah mendapatkan nilai AMB, selanjutnya dapat dihitung kalori berdasarkan faktor-faktor yang mempengaruhinya antara lain:

- Berat badan

Berdasarkan hasil perhitungan berat badan ideal maka ibu hamil tersebut masuk ke dalam kategori normal sehingga dapat dihitung kalori berat menggunakan Persamaan (2.3):

$$\text{Kalori berat} = 0\% \times 1260 = 0 \text{ kkal}$$

- Usia

Pengaruh usia terhadap kebutuhan kalori ibu hamil dapat dilihat pada Tabel 2.3. Kalori usia dapat dihitung menggunakan Persamaan (2.4):

$$\text{Kalori usia} = 0 \times 1260 = 0 \text{ kkal}$$

- Aktivitas fisik

Berdasarkan jenis aktivitas pada permasalahan didapatkan nilai aktivitas fisik berdasarkan pada Tabel 2.4. Kalori aktivitas dapat dihitung menggunakan Persamaan (2.5)

$$\text{Kalori aktivitas} = 20\% \times 1260 = 252 \text{ kkal}$$

- Usia kehamilan

Usia kehamilan memberikan tambahan energi yang berbeda-beda pada ibu hamil seperti ditunjukkan pada Tabel 2.5. Pada contoh permasalahan usia kehamilan adalah 8 minggu sehingga mendapatkan tambahan energi sebesar 180 kkal.

Langkah terakhir dalam menentukan kebutuhan energi yaitu menghitung total kalori harian yang dibutuhkan oleh ibu hamil dengan menggunakan Persamaan (2.6).

$$Total\ Kalori = 1260 + 0 + 0 + 252 + 0 + 180 = 1692\ kkal$$

Setelah mendapatkan nilai kalori harian yang dibutuhkan selanjutnya menghitung kebutuhan gizi yang dapat dikonsumsi sebagai berikut:

- $Karbohidrat = \left(\frac{65\% \times 1692}{4} \right) + 25 = 299.95\ gr$
- $Protein = \left(\frac{15\% \times 1692}{4} \right) + 20 = 83.45\ gr$
- $Lemak = \left(\frac{20\% \times 1692}{9} \right) + 6 = 43.6\ gr$
- $Natrium = 1000 - 1200\ mg$

4.4.2 Inisialisasi parameter algoritme genetika

Pada algoritme genetika terdapat parameter-parameter yang harus didefinisikan terlebih dahulu antara lain, jumlah generasi, *popsi*, *crossover rate* (*cr*), dan *mutation rate* (*mr*). Untuk menyelesaikan permasalahan pada contoh permasalahan ditentukan parameter algoritme genetika sebagai berikut:

- Jumlah generasi : 2
- *Popsi* : 3
- *Crossover rate* (*cr*) : 0.6
- *Mutation rate* (*mr*) : 0.3

4.4.3 Representasi kromosom

Pada penelitian ini digunakan 110 makanan olahan yang dibagi menjadi 5 jenis yaitu sumber karbohidrat sebanyak 22 jenis, protein hewani sebanyak 25 jenis, protein nabati sebanyak 17 jenis, sayuran sebanyak 23 jenis dan buah-buahan sebanyak 23 jenis. Representasi kromosom yang digunakan adalah representasi bilangan integer. Bilangan tersebut menunjukkan id makanan yang akan dikonsumsi.

Berdasarkan contoh permasalahan akan dilakukan optimasi asupan makanan untuk tiga hari sehingga panjang kromosom yang digunakan sebesar 45 yang terbagi dalam 3 kali makan dalam satu hari yaitu makan pagi, makan siang dan makan malam. Setiap kelipatan 15 pada gen kromosom tersebut menunjukkan hari, dan setiap 5 gen menunjukkan menu makanan yang dikonsumsi pada satu kali makan yang terdiri dari sumber karbohidrat, protein hewani, protein nabati, sayuran, dan buah-buahan.

Contoh representasi kromosom yang digunakan dalam menyelesaikan contoh permasalahan ditunjukkan pada Tabel 4.2.

Tabel 4.2 Contoh representasi kromosom

Individu	Hari Ke-	Pagi					Siang					Malam				
		K	Ph	Pn	S	B	K	Ph	Pn	S	B	K	Ph	Pn	S	B
P1	1	5	21	7	9	5	17	15	12	18	7	6	14	16	2	15
	2	4	18	12	2	17	18	16	15	3	7	1	13	1	1	20
	3	12	4	7	22	9	7	23	8	12	4	5	23	2	20	20
P2	1	9	1	9	15	9	9	9	12	9	2	15	1	3	1	4
	2	7	22	1	9	4	11	12	7	13	15	7	6	10	3	12
	3	2	18	15	13	8	12	5	11	14	1	12	6	6	1	21
P3	1	4	2	4	4	15	17	11	3	7	17	5	23	1	17	16
	2	7	3	2	5	1	5	10	1	14	20	6	10	12	12	9
	3	11	6	1	9	19	12	12	5	11	8	9	9	8	2	9

Contoh representasi kromosom pada Tabel 4.2 dapat dikonversikan menjadi daftar menu makanan yang ditunjukkan pada Tabel 4.3.

Tabel 4.3 Konversi kromosom menjadi menu makanan

<i>P1</i>								
Hari ke-	Waktu makan	Nama makanan	Kalori (kkal)	Karbohidrat (gr)	Protein (gr)	Lemak (gr)	Natrium (mg)	Harga (Rp)
1	Pagi	Nasi uduk	212.5	42.3	3.8	2.7	1.8	3200
		Hati ayam goreng	157.0	0.4	12.2	2.8	25.5	2500
		Semur tahu	137.0	10.6	13.7	4.5	26.0	1500
		Tumis bayam dan sawi	187.9	2.4	2.4	20.2	22.0	4200
		Belimbing	64.1	14.4	1.2	0.8	4.0	3800
	Siang	Mie rebus	253.8	50.9	8.6	1.3	1.8	2000
		Telur asin	77.6	0.6	6.3	5.3	62.0	2500
		Sup kacang hijau	14.5	2.6	0.9	0.1	0.5	2000
		Sayur nangka	131.9	17.4	2.6	7.0	8.0	5500
		Jeruk manis	94.2	23.6	1.8	0.2	0.0	5000
	Malam	Bubur nasi	131.2	28.8	2.3	0.2	0.0	2000
		Ayam rebus	128.3	0.0	9.5	10.1	50.0	3500
		Sayur tempe	132.5	8.4	7.9	8.5	3.5	1700
		Sayur bayam	23.9	3.8	3.0	0.4	70.0	2500
2	Pagi	Nasi tim	210.8	46.3	4.0	0.4	0.0	3300
		Telur puyuh	92.5	0.8	6.4	6.9	73.0	2000
		Sup kacang hijau	14.5	2.6	0.9	0.1	0.5	2000

Tabel 4.3 Konversi kromosom menjadi menu makanan (lanjutan)

P1								
Hari ke-	Waktu makan	Nama makanan	Kalori (kkal)	Karbohidrat (gr)	Protein (gr)	Lemak (gr)	Natrium (mg)	Harga (Rp)
3		Sayur bayam	23.9	3.8	3.0	0.4	70.0	2500
		Strawberry	64.1	11.0	1.6	0.8	6.0	3500
	Siang	Mie kering	585.1	101.9	17.3	11.5	3.6	2000
		Telur ceplok	95.5	0.5	6.0	7.6	59.0	2000
		Lodeh tahu dan tempe	82.9	4.8	4.9	5.6	6.0	2500
		Sayur kangkung	22.0	3.0	3.2	0.2	22.0	5000
		Jeruk manis	94.2	23.6	1.8	0.2	0.0	5000
	Malam	Nasi putih	234.0	51.5	4.3	0.4	0.0	3000
		Ikan asin goreng	81.0	0.0	6.5	6.0	17.0	1000
		Tempe goreng	177.0	7.7	8.6	13.4	2.5	1000
		Sayur asem	98.0	11.4	3.6	5.4	10.0	3500
		Salak	164.0	42.6	0.4	0.8	0.0	1550
	Pagi	Bihun goreng	426.8	41.2	3.4	27.4	0.0	2400
		Udang goreng	46.6	0.4	9.4	0.8	73.0	7500
		Semur tahu	137.0	10.6	13.7	4.5	26.0	1500
		Ketimun mentah	25.8	5.6	1.4	0.2	4.0	2500
		Mangga gadung	130.0	34.0	1.0	0.6	4.0	3000
	Siang	Bubur tim	142.4	29.0	3.6	0.9	10.8	2200
		Semur daging ayam	229.4	0.0	5.4	23.8	14.5	6000
		Pepes jamur	13.5	2.5	1.1	0.3	1.0	2500
		Tumis selada air	390.0	1.6	0.8	44.2	8.0	4000
		Arbei	104.2	25.6	1.4	0.8	0.0	2750
	Malam	Nasi uduk	212.5	42.3	3.8	2.7	1.8	3200

Tabel 4.3 Konversi kromosom menjadi menu makanan (lanjutan)

P1								
Hari ke-	Waktu makan	Nama makanan	Kalori (kkal)	Karbohidrat (gr)	Protein (gr)	Lemak (gr)	Natrium (mg)	Harga (Rp)
		Semur daging ayam	229.4	0.0	5.4	23.8	14.5	6000
		Tempe bacem	118.5	8.8	5.4	7.5	2.5	1200
		Bayam merah	74.1	14.6	7.4	0.4	22.0	7250
		Salak	164.0	42.6	0.4	0.8	0.0	1550

4.4.4 Inisialisasi populasi awal

Berdasarkan contoh permasalahan ditentukan jumlah *popsiz*e yaitu 3 dan jumlah hari adalah 3 sehingga panjang kromosom setiap individu adalah $3 \times 15 = 45$. Inisialisasi populasi awal pada contoh permasalahan ditunjukkan pada Tabel 4.4 dengan P1, P2 dan P3 merupakan individu-individu yang terbentuk dalam populasi awal.

Tabel 4.4 Inisialisasi populasi awal

Individu	Kromosom														
P1	5	21	7	9	5	17	15	12	18	7	6	14	16	2	15
	4	18	12	2	17	18	16	15	3	7	1	13	1	1	20
	12	4	7	22	9	7	23	8	12	4	5	23	2	20	20
P2	9	1	9	15	9	9	9	12	9	2	15	1	3	1	4
	7	22	1	9	4	11	12	7	13	15	7	6	10	3	12
	2	18	15	13	8	12	5	11	14	1	12	6	6	1	21
P3	4	2	4	4	15	17	11	3	7	17	5	23	1	17	16
	7	3	2	5	1	5	10	1	14	20	6	10	12	12	9
	11	6	1	9	19	12	12	5	11	8	9	9	8	2	9

4.4.5 Crossover

Proses *crossover* pada permasalahan ini menggunakan *extended intermediate crossover*. Awalnya ditentukan terlebih dahulu jumlah *offspring* yang akan dihasilkan dengan cara mengalikan nilai *crossover rate* (*cr*) dengan *popsiz*e. Pada permasalahan ini dapat dihitung nilai *offspring* sebagai berikut:

$$offspring = 0.6 \times 3 = 1.8 \text{ (dibulatkan menjadi 2)}$$

Berdasarkan perhitungan di atas proses *crossover* akan menghasilkan 2 *offspring* sehingga hanya memerlukan satu kali proses *crossover*. Langkah selanjutnya yaitu memilih dua induk dalam populasi awal secara acak. Dimisalkan 2 induk yang dipilih yaitu P1 dan P2 dengan interval nilai $\alpha = [0.1 ; 1]$ maka proses *crossover* dapat dilihat pada Tabel 4.5.

Tabel 4.5 Proses crossover

Individu	Kromosom														
P1	5	21	7	9	5	17	15	12	18	7	6	14	16	2	15
	4	18	12	2	17	18	16	15	3	7	1	13	1	1	20
	12	4	7	22	9	7	23	8	12	4	5	23	2	20	20
α	0.3	0.5	0.6	0.9	0.1	0.3	0.6	0.3	0.6	0.7	0.6	0.5	0.1	0.2	0.1
	0.2	0.9	0.4	0.1	0.7	0.5	0.6	0.1	0.8	0.4	0.5	0.1	0.1	0.1	0.3
	0.5	0.3	0.2	0.3	0.8	0.7	0.6	0.1	0.5	0.6	1.0	0.9	0.5	0.5	0.4
P2	9	1	9	15	9	9	9	12	9	2	15	1	3	1	4
	7	22	1	9	4	11	12	7	13	15	7	6	10	3	12
	2	18	15	13	8	12	5	11	14	1	12	6	6	1	21

Dengan menggunakan Persamaan (2.4) dan (2.5) maka didapatkan *offspring* C1 dan C2 yang ditunjukkan pada Tabel 4.7.

$$C1 \text{ gen ke } - 1 = 5 + 0.3(9 - 5) = 6.2 \text{ (dibulatkan menjadi 6)}$$

$$C2 \text{ gen ke } - 1 = 9 + 0.3(5 - 9) = 7.8 \text{ (dibulatkan menjadi 8)}$$

Dari proses *crossover* didapatkan dua buah *offspring* yaitu C1 dan C2 yang dapat dilihat pada Tabel 4.6.

Tabel 4.6 Offspring hasil crossover

Individu	Kromosom														
C1	6	11	8	14	5	15	11	12	13	4	11	8	15	2	14
	5	22	8	3	8	15	14	14	11	10	4	12	2	1	18
	7	8	9	19	8	11	12	8	13	2	12	8	4	11	20
C2	8	11	8	10	9	11	13	12	14	6	10	8	4	1	5
	6	18	5	8	13	15	14	8	5	12	4	7	9	3	14
	7	14	13	16	9	9	16	11	13	3	5	21	4	11	21

4.4.6 Mutasi

Metode mutasi yang digunakan yaitu *random mutation*. Proses ini diawali dengan menentukan jumlah *offspring* yang akan dihasilkan dengan cara mengalikan nilai *mutation rate* (*mr*) dengan *popsiz*e sebagai berikut:

$$offspring = 0.3 \times 3 = 0.9 \text{ (dibulatkan menjadi 1)}$$

Berdasarkan hasil perhitungan maka hanya membutuhkan satu kali proses mutasi. Langkah selanjutnya yaitu memilih induk dari populasi awal secara acak. Dimisalkan induk terpilih pada proses mutasi adalah P3 dan ditentukan nilai $r = [0,1 ; 1]$. Induk mutasi dapat dilihat pada Tabel 4.7.

Tabel 4.7 Induk mutasi

Individu	Kromosom														
P3	4	2	4	4	15	17	11	3	7	17	5	23	1	17	16
	7	3	2	5	1	5	10	1	14	20	6	10	12	12	9
	11	6	1	9	19	12	12	5	11	8	9	9	8	2	9

Sehingga dapat dihitung hasil mutasi C3 dengan nilai $r = 0,3$ menggunakan Persamaan (2.9) sebagai berikut:

$$C3 \text{ gen ke} - 16 = 7 + (0,3 \times (23 - 1)) = 13,6 \text{ (dibulatkan menjadi 14)}$$

Dari proses mutasi didapatkan satu *offspring* yaitu C3 yang dapat dilihat pada Tabel 4.8.

Tabel 4.8 Offspring hasil mutasi

Individu	Kromosom														
C3	11	9	11	11	22	22	18	10	14	23	12	25	8	23	23
	14	10	9	12	8	12	17	8	21	23	13	17	19	19	16
	18	13	8	16	23	19	19	12	18	15	16	16	15	9	16

4.4.7 Evaluasi

Setelah melalui proses reproduksi yaitu *crossover* dan mutasi didapatkan populasi baru yaitu populasi awal dan populasi *offspring* hasil *crossover* dan mutasi seperti ditunjukkan pada Tabel 4.9.

Tabel 4.9 Populasi baru

Individu	Kromosom														
P1	5	21	7	9	5	17	15	12	18	7	6	14	16	2	15
	4	18	12	2	17	18	16	15	3	7	1	13	1	1	20
	12	4	7	22	9	7	23	8	12	4	5	23	2	20	20
P2	9	1	9	15	9	9	9	12	9	2	15	1	3	1	4
	7	22	1	9	4	11	12	7	13	15	7	6	10	3	12
	2	18	15	13	8	12	5	11	14	1	12	6	6	1	21
P3	4	2	4	4	15	17	11	3	7	17	5	23	1	17	16
	7	3	2	5	1	5	10	1	14	20	6	10	12	12	9
	11	6	1	9	19	12	12	5	11	8	9	9	8	2	9
C1	6	11	8	14	5	15	11	12	13	4	11	8	15	2	14
	5	22	8	3	8	15	14	14	11	10	4	12	2	1	18
	7	8	9	19	8	11	12	8	13	2	12	8	4	11	20
C2	8	11	8	10	9	11	13	12	14	6	10	8	4	1	5
	6	18	5	8	13	15	14	8	5	12	4	7	9	3	14
	7	14	13	16	9	9	16	11	13	3	5	21	4	11	21
C3	11	9	11	11	22	22	18	10	14	23	12	25	8	23	23
	14	10	9	12	8	12	17	8	21	23	13	17	19	19	16
	18	13	8	16	23	19	19	12	18	15	16	16	15	9	16

Pada proses evaluasi akan dihitung nilai *fitness* masing-masing individu dalam populasi untuk menentukan kualitas masing-masing individu. Sebelum menghitung nilai *fitness* dihitung terlebih dahulu penalti masing-masing gizi.

4.4.7.1 Perhitungan penalti

Setelah melalui proses *crossover* dan mutasi populasi akan bertambah dengan *offspring* hasil *crossover* dan mutasi seperti ditunjukkan pada Tabel 4.9. Langkah pertama yang harus dilakukan dalam menghitung penalti masing-masing gizi adalah menghitung total gizi yaitu karbohidrat, protein, lemak, dan natrium

masing-masing individu. Jumlah kandungan gizi sistem ditunjukkan pada Tabel 4.10.

Tabel 4.10 Jumlah kandungan gizi makanan sistem

<i>P1</i>						
Hari ke-	Total					
	Kalori	Karbohidrat	Protein	Lemak	Natrium	Harga
1	1978.2	268.6	77.8	64.6	285.1	43100
2	2039.5	311.5	72.5	59.7	269.6	39850
3	2257.4	271.4	78.8	103.5	224.1	49550

Berdasarkan jumlah kandungan gizi makanan pada Tabel 4.10 dapat dihitung penalti masing-masing kebutuhan gizi menggunakan Persamaan (2.13) sampai (2.17) sebagai berikut:

- Hari ke-1

$$\text{Penalti kalori} = |1692 - 1978.2| = 286.2 \text{ kkal}$$

$$\text{Penalti karbohidrat} = |299.95 - 268.6| = 31.35 \text{ gr}$$

$$\text{Penalti protein} = |83.45 - 77.8| = 5.65 \text{ gr}$$

$$\text{Penalti lemak} = |43.6 - 64.6| = 21 \text{ gr}$$

$$\text{Penalti natrium} = |1000 - 285.1| = 714.9 \text{ mg}$$

- Hari ke-2

$$\text{Penalti kalori} = |1692 - 2039.5| = 347.5 \text{ kkal}$$

$$\text{Penalti karbohidrat} = |299.95 - 311.5| = 11.55 \text{ gr}$$

$$\text{Penalti protein} = |83.45 - 72.5| = 10.95 \text{ gr}$$

$$\text{Penalti lemak} = |43.6 - 59.7| = 16.1 \text{ gr}$$

$$\text{Penalti natrium} = |1000 - 269.6| = 730.4 \text{ mg}$$

- Hari ke-3

$$\text{Penalti kalori} = |1692 - 2257.4| = 565.4 \text{ kkal}$$

$$\text{Penalti karbohidrat} = |299.95 - 271.4| = 28.55 \text{ gr}$$

$$\text{Penalti protein} = |83.45 - 78.8| = 4.65 \text{ gr}$$

$$\text{Penalti lemak} = |43.6 - 103.5| = 59.9 \text{ gr}$$

$$\text{Penalti natrium} = |1000 - 224.1| = 775.9 \text{ mg}$$

Setelah mendapatkan nilai penalti masing-masing kebutuhan gizi, selanjutnya adalah menghitung penalti variasi makanan. Jika terdapat jenis makanan dengan nama yang sama pada hari yang sama maka nilai variasi adalah 0 dan jika tidak ada yang sama maka variasi bernilai 1. Perhitungan penalti variasi dapat dilihat pada Tabel 4.11.

Tabel 4.11 Perhitungan variasi

<i>P1</i>															
Hari Ke-	Pagi					Siang					Malam				
	K	Ph	Pn	S	B	K	Ph	Pn	S	B	K	Ph	Pn	S	B
1	5	21	7	9	5	17	15	12	18	7	6	14	16	2	15
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	4	18	12	2	17	18	16	15	3	7	1	13	1	1	20
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	12	4	7	22	9	7	23	8	12	4	5	23	2	20	20
	1	1	1	1	1	1	0	1	1	1	1	0	1	1	1

Pada tabel 4.11 dapat dilihat protein nabati hari ke-1 pada makan pagi, siang dan malam tidak ada yang sama maka diberikan nilai variasi adalah 1, sedangkan pada protein hewani pada makan siang dan malam hari ke-3 sama maka diberi nilai nol. Berdasarkan hasil diatas jumlah makanan yang berbeda adalah 43.

4.4.7.2 Perhitungan fitness

Setelah mendapatkan penalti masing-masing gizi dan penalti variasi, selanjutnya menghitung nilai *fitness* masing-masing individu menggunakan Persamaan (2.12). Hasil perhitungan *fitness* dapat dilihat pada Tabel 4.12.

$$\begin{aligned}
 \text{Fitness } P1 &= \frac{1}{1 + \frac{1199.10}{100}} + \frac{1}{1 + 71.45} + \frac{50}{1 + 21.25} + \frac{20}{1 + 97} + \frac{50}{1 + 2221.20} \\
 &\quad + \frac{100}{132500} + (5 \times 43) \\
 &= 217.4892
 \end{aligned}$$

Tabel 4.12 Hasil perhitungan *fitness*

Individu	Total Penalti					Total Harga (Rp)	Variasi	Fitness
	Kalori (kkal)	Karbohidrat (gr)	Protein (gr)	Lemak (gr)	Natrium (mg)			
<i>P1</i>	1199.10	71.45	21.25	97.00	2221.20	132500.00	43	217.4892
<i>P2</i>	1044.00	146.65	34.65	150.80	2022.80	134600.00	37	186.5675
<i>P3</i>	1121.20	138.35	35.55	117.10	2378.20	122150.00	43	216.5672
<i>C1</i>	700.60	162.75	71.85	80.30	1719.90	128900.00	41	205.9697
<i>C2</i>	716.20	200.35	76.25	72.10	1947.40	130050.00	45	225.9536
<i>C3</i>	1773.10	130.95	70.95	162.20	1637.50	122300.00	41	205.8570

4.4.8 Seleksi

Proses seleksi pada penyelesaian permasalahan ini menggunakan metode seleksi *elitism*. Individu-individu dalam populasi diurutkan berdasarkan nilai *fitness* terbesar hingga terkecil dan individu sebanyak *popsiz*e dengan nilai *fitness* terbesar akan terpilih. Hasil proses seleksi ditunjukkan pada Tabel 4.13

Tabel 4.13 Hasil seleksi

Individu	Kromosom															Fitness
C2	8	11	8	10	9	11	13	12	14	6	10	8	4	1	5	225.9536
	6	18	5	8	13	15	14	8	5	12	4	7	9	3	14	
	7	14	13	16	9	9	16	11	13	3	5	21	4	11	21	
P1	5	21	7	9	5	17	15	12	18	7	6	14	16	2	15	217.4892
	4	18	12	2	17	18	16	15	3	7	1	13	1	1	20	
	12	4	7	22	9	7	23	8	12	4	5	23	2	20	20	
P3	4	2	4	4	15	17	11	3	7	17	5	23	1	17	16	216.5672
	7	3	2	5	1	5	10	1	14	20	6	10	12	12	9	
	11	6	1	9	19	12	12	5	11	8	9	9	8	2	9	

Populasi baru tersebut akan melalui tahap *crossover*, mutasi, evaluasi dan seleksi pada generasi ke-2. Hasil perhitungan *fitness* pada generasi ke-2 dapat dilihat pada Tabel 4.14

Tabel 4.14 Hasil perhitungan *fitness* generasi ke-2

Individu	Total Penalti					Total Harga (Rp)	Variasi	Fitness
	Kalori (kkal)	Karbohidrat (gr)	Protein (gr)	Lemak (gr)	Natrium (mg)			
P1	716.20	200.35	76.25	72.10	1947.40	130050.00	45	225.9536
P2	1199.10	71.45	21.25	97.00	2221.20	132500.00	43	217.4892
P3	1121.20	138.35	35.55	117.10	2378.20	122150.00	43	216.5672
C1	815.90	209.45	65.15	26.00	1923.10	133250.00	39	196.5293
C2	1773.10	130.95	70.95	162.20	1637.50	122300.00	43	215.8570
C3	732.80	167.65	64.75	56.20	2350.00	146450.00	41	206.1393

Pada generasi ke-2 individu terbaik yang terpilih untuk dijadikan solusi adalah P1 dengan nilai *fitness* 225,9536

4.5 Perancangan antarmuka

Antarmuka digunakan sebagai perantara pengguna untuk dapat berinteraksi dengan sistem . Perancangan antarmuka pada sistem ini terdiri dari halaman *input* data, halaman proses algoritme genetika, halaman data makanan, dan halaman hasil menu makanan.

4.5.1 Perancangan antarmuka halaman utama

Antarmuka halaman utama terdiri atas *input* data ibu hamil, *input* operator algoritme genetika dan hasil perhitungan kebutuhan energi dan gizi harian ibu hamil. Perancangan halaman utama ditunjukkan pada Gambar 4.9.

SISTEM OPTIMASI ASUPAN MAKANAN HARIAN IBU HAMIL PENDERITA HIPERTENSI

Data Input

Nama

Usia

Usia Kehamilan minggu

Berat Badan kg

Tinggi Badan cm

Tekanan Darah mmHg

Aktivitas

Jumlah Hari

Operator Algoritme Genetika

Generasi

PopSize

Crossover Rate

Mutation Rate

Default

Submit

Kebutuhan Gizi | Proses Algoritma Genetika | Hasil Menu Makanan | Daftar Makanan

Nama

Usia	Usia Kehamilan	Berat Badan	Tinggi Badan	Tekanan Darah	Aktivitas

Kebutuhan Gizi Harian

Kebutuhan Kalori	<input type="text"/>	kcal
Kebutuhan Karbohidrat	<input type="text"/>	gram
Kebutuhan Protein	<input type="text"/>	gram
Kebutuhan Lemak	<input type="text"/>	gram
Kebutuhan Natrium	<input type="text"/>	mg

Gambar 4.9 Perancangan antarmuka halaman utama

4.5.2 Perancangan antarmuka halaman proses algoritme genetika

Halaman proses algoritme genetika akan menunjukkan bagaimana hasil setiap proses pada algoritme genetika di setiap generasi. Perancangan halaman proses algoritme genetika ditunjukkan pada Gambar 4.10.

SISTEM OPTIMASI ASUPAN MAKANAN HARIAN IBU HAMIL PENDERITA HIPERTENSI

Data Input

Nama

Usia

Usia Kehamilan minggu

Berat Badan kg

Tinggi Badan cm

Tekanan Darah mmHg

Aktivitas

Jumlah Hari

Operator Algoritme Genetika

Generasi

PopSize

Crossover Rate

Mutation Rate

Default

Submit

Kebutuhan Gizi | Proses Algoritma Genetika | Hasil Menu Makanan | Daftar Makanan

Populasi Awal | Crossover | Mutasi | Individu Terbaik

Generasi ke-	Individu	Kromosom

Gambar 4.10 Perancangan antarmuka halaman proses algoritme genetika

4.5.3 Perancangan antramuka halaman hasil optimasi

Pada halaman hasil optimasi menu makanan ini akan menunjukkan daftar rekomendasi menu makanan yang dapat dikonsumsi sejumlah hari yang ditentukan. Perancangan halaman hasil optimasi menu makanan ditunjukkan pada Gambar 4.11.

Gambar 4.11 Perancangan antarmuka halaman hasil menu makanan

Pada halaman ini akan menampilkan seluruh daftar makanan yang digunakan pada sistem ini. Perancangan halaman daftar makanan ditunjukkan pada Gambar 4.12.

Gambar 4.12 Perancangan antarmuka halaman daftar makanan

Pengujian yang akan dilakukan terhadap sistem ini terdiri dari 3 pengujian antara lain:

1. Pengujian generasi atau iterasi maksimum
2. Pengujian populasi yaitu menentukan ukuran *popsize*

3. Pegujian dengan kombinasi nilai *crossover rate* dan *mutation rate*

4.6.1 Pengujian ukuran generasi

Pengujian ukuran generasi pada sistem ini dilakukan untuk mengetahui berapa ukuran generasi optimal yang dapat digunakan sisten untuk menghasilkan hasil optimal. Semakin tinggi ukuran dan kompleksitas masalah maka ukuran generasi yang dibutuhkan semakin besar. Pengujian generasi pada sistem ini dilakukan sebanyak 10 kali pada setiap generasi. Uji coba generasi dapat dilihat pada Tabel 4.15.

Tabel 4.15 Pengujian ukuran generasi

Generasi	Nilai <i>Fitness</i> Pengujian Ke-										Rata-rata Nilai <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
30											
60											
90											
120											
150											
180											
210											
240											
270											
300											

4.6.2 Pengujian ukuran populasi

Pengujian ukuran populasi dilakukan dengan menentukan nilai *popsiz* yaitu banyaknya individu dalam populasi. Pengujian ini dilakukan untuk menentukan pengaruh ukuran populasi terhadap besarnya nilai *fitness* yang dihasilkan. Pengujian ukuran populasi pada sistem ini dilakukan sebanyak 10 kali dengan pada setiap ukuran populasi. Semakin besar ukuran populasi maka semakin banyak dan beragam individu yang dapat menghasilkan solusi terbaik. Uji coba ukuran populasi dapat dilihat pada Tabel 4.16.

Tabel 4.16 Pengujian ukuran populasi

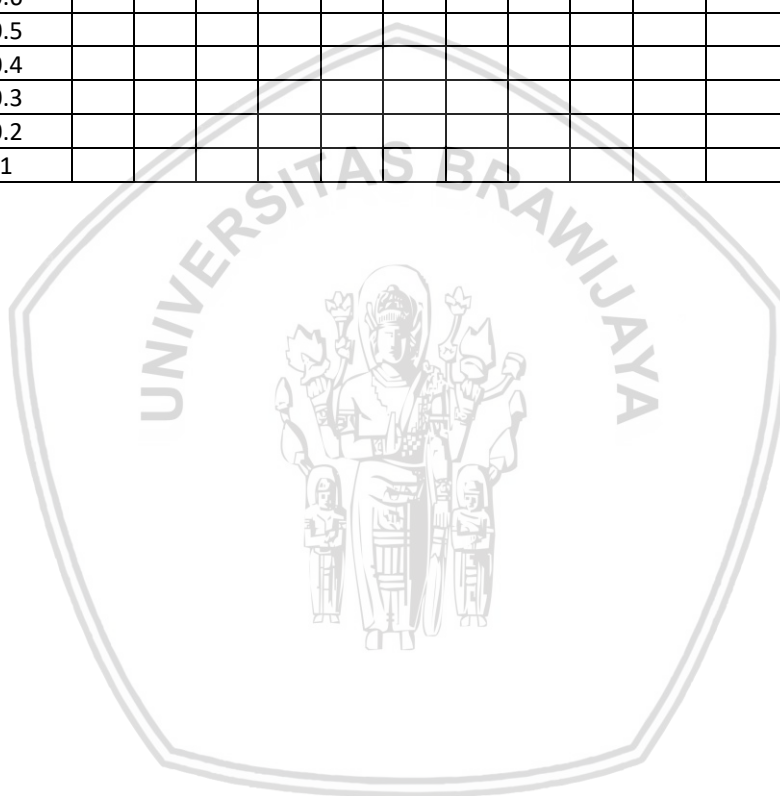
Populasi	Nilai <i>Fitness</i> Pengujian Ke-										Rata-rata Nilai <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
10											
20											
30											
40											
50											
60											
70											
80											
90											
100											
110											

4.6.3 Pengujian kombinasi *crossover rate* dan *mutation rate*

Pengujian kombinasi nilai *crossover rate* (cr) dan *mutation rate* menggunakan bilangan antara 0.1 hingga 1. Uji coba setiap kombinasi nilai cr dan mr dilakukan sebanyak 10 kali. Uji coba tersebut dapat dilihat pada Tabel 4.17.

Tabel 4.17 Pengujian cr dan mr

$C_r ; M_r$	Nilai <i>Fitness</i> Pengujian Ke-										Rata-rata Nilai <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
0.1 ; 1											
0.2 ; 0.9											
0.3 ; 0.8											
0.4 ; 0.7											
0.5 ; 0.6											
0.6 ; 0.5											
0.7 ; 0.4											
0.8 ; 0.3											
0.9 ; 0.2											
1 ; 0.1											



BAB 5 IMPLEMENTASI

Bab ini menjelaskan tentang implementasi sistem yang berdasarkan analisis kebutuhan serta proses perancangan sistem sebelumnya. Pada bab ini juga menjelaskan tentang implementasi dari algoritme genetika dan antarmuka yang digunakan dalam menyelesaikan permasalahan optimasi asupan makanan harian ibu hamil penderita Hipertensi.

5.1 Spesifikasi sistem

Dalam proses implementasi sistem terdapat spesifikasi perangkat yang digunakan dalam membangun sistem. Spesifikasi perangkat tersebut terdiri dari spesifikasi perangkat keras dan spesifikasi perangkat lunak.

5.1.1 Spesifikasi perangkat keras

Spesifikasi perangkat keras yang digunakan dalam implementasi algoritme genetika untuk optimasi asupan makanan harian ibu hamil penderita Hipertensi ditunjukkan pada Tabel 5.1.

Tabel 5.1 Spesifikasi perangkat keras

Komponen	Spesifikasi
Prosesor	Intell Corel i7-3610QM CPU @ 2.30GHz 2.30GHz
Memori	2048 MB
Hardisk	500 GB

5.1.2 Spesifikasi perangkat lunak

Spesifikasi perangkat lunak yang digunakan dalam implementasi algoritme genetika untuk optimasi asupan makanan harian ibu hamil penderita Hipertensi ditunjukkan pada Tabel 5.2.

Tabel 5.2 Spesifikasi perangkat lunak

Komponen	Spesifikasi
Sistem operasi	Windows 10
Bahasa pemrograman	Java
Tools pemrograman	NetBeans IDE

5.2 Implementasi program

Implementasi program menjelaskan tentang bagaimana proses algoritme genetika dalam menyelesaikan masalah optimasi asupan makanan harian ibu hamil penderita Hipertensi dalam bentuk kode program.

5.2.1 Implementasi proses inisialisasi populasi awal

Pada proses inisialisasi populasi awal sistem akan membangkitkan populasi secara acak sebanyak nilai *popsiz*e yang telah ditentukan. Panjang kromosom ditentukan oleh jumlah hari dikalikan dengan 15 yaitu panjang kromosom dalam satu hari makan. Proses inisialisasi populasi awal ditunjukkan pada Kode Program 5.1.

```

1  for (int i = 0; i < popsize; i++) {
2      for (int j = 0; j < panjangkromosom; j++) {
3          if (j % 5 == 0) {
4              populasi[i][j] = Math.round((1 + (int)
5                  (Math.random() * 100)) % 23);
6          }
7          if (j % 5 == 1) {
8              populasi[i][j] = Math.round((1 + (int)
9                  (Math.random() * 100)) % 26);
10         }
11         if (j % 5 == 2) {
12             populasi[i][j] = Math.round((1 + (int)
13                 (Math.random() * 100)) % 18);
14         }
15         if (j % 5 == 3) {
16             populasi[i][j] = Math.round((1 + (int)
17                 (Math.random() * 100)) % 24);
18         }
19         if (j % 5 == 4) {
20             populasi[i][j] = Math.round((1 + (int)
21                 (Math.random() * 100)) % 24);
22         }
23     }
24 }

```

Kode Program 5.1 Implementasi proses inisialisasi populasi awal

Berikut penjelasan Kode Program 5.1:

1. Baris 1-2, perulangan untuk pembangkitan populasi sebanyak *popsiz*e dengan jumlah gen sebesar panjangkromosom.
2. Baris 3-5, proses menentukan gen jika index mod 5 = 0 maka gen tersebut merupakan jenis sumber karbohidrat.
3. Baris 6-8, proses menentukan gen jika index mod 5 = 1 maka gen tersebut merupakan jenis protein hewani.
4. Baris 9-11, proses menentukan gen jika index mod 5 = 2 maka gen tersebut merupakan jenis protein nabati.
5. Baris 12-14, proses menentukan gen jika index mod 5 = 3 maka gen tersebut merupakan jenis sayuran.
6. Baris 15-17, proses menentukan gen jika index mod 5 = 4 maka gen tersebut merupakan jenis buah-buahan.

5.2.2 Implementasi proses *crossover*

Proses *crossover* merupakan proses reproduksi pertama yang dilakukan untuk menghasilkan keturunan baru. Diawali dengan menghitung *offspring* yang dihasilkan yaitu mengalikan nilai *cr* dengan nilai *popsiz*, kemudian memilih dua buah induk secara acak untuk melalui proses *crossover*. Dalam satu kali proses *crossover* akan menghasilkan 2 individu baru. Proses *crossover* ditunjukkan pada Kode Program 5.2.

```

1  for (int i = 0; i < child; i = i + 2) {
2      boolean sama = true;
3      P[0] = r.nextInt(popsiz - 1);
4      while (sama) {
5          P[1] = r.nextInt(popsiz - 1);
6          if (P[0] != P[1]) {
7              sama = false;
8          }
9      }
10     for (int j = 0; j < panjangkromosom; j++) {
11         indukc[0][j] = populasi[P[0]][j];
12     }
13     for (int j = 0; j < panjangkromosom; j++) {
14         indukc[1][j] = populasi[P[1]][j];
15     }
16     for (int j = 0; j < panjangkromosom; j++) {
17         angka = 0.1 + (double) (Math.random() * 1);
18         alpha[j] = angka;
19     }
20     for (int k = 0; k < panjangkromosom; k++) {
21         hasilcrossovertemp[0][k] = (int)
22             Math.round(indukc[0][k] + (alpha[k] *
23                 (indukc[1][k] - indukc[0][k])));
24         hasilcrossovertemp[1][k] = (int)
25             Math.round(indukc[1][k] + (alpha[k] *
26                 (indukc[0][k] - indukc[1][k])));
27     }
28     for (int j = 0; j < panjangkromosom; j++) {
29         if (child % 2 != 0 && i == child - 1) {
30             hasilcrossover[i][j] =
31                 hasilcrossovertemp[0][j];
32         } else {
33             hasilcrossover[i][j] =
34                 hasilcrossovertemp[0][j];
35             hasilcrossover[i + 1][j] =
36                 hasilcrossovertemp[1][j];
37         }
38     }
39 }

```

Kode Program 5.2 Implementasi proses *crossover*

Berikut penjelasan Kode Program 5.2:

1. Baris 2-9, proses memilih dua individu acak.
2. Baris 10-15, proses inisialisasi induk *crossover*.
3. Baris 16-19, proses menentukan nilai *alpha* secara acak.

4. Baris 20-23, proses *extended intermediate crossover*.
5. Baris 24-32, proses menampung hasil *crossover*.

5.2.3 Implementasi proses mutasi

Proses reproduksi kedua yang dilakukan adalah proses mutasi. Pada proses ini ditentukan terlebih dahulu *offspring* yang dihasilkan yaitu mengalikan nilai *mr* dengan nilai *popsiz*. Kemudian memilih parent secara acak untuk dilakukan mutasi, satu kali proses mutasi akan menghasilkan satu individu baru. Implementasi proses mutasi ditunjukkan pada Kode Program 5.3.

```

1  for (int i = 0; i < child; i++) {
2      int p = r.nextInt(popsiz - 1);
3      for (int j = 0; j < panjangkromosom; j++) {
4          induk[0][j] = populasi[p][j];
5      }
6      for (int k = 0; k < panjangkromosom; k++) {
7          if (induk[0][k] < min) {
8              min = induk[0][k];
9          }
10         if (induk[0][k] > max) {
11             max = induk[0][k];
12         }
13     }
14     int x = max - min;
15     double r = 0.1 + (double) (Math.random() * 1);
16     for (int k = 0; k < panjangkromosom; k++) {
17         hasilmutasitemp[0][k] = (int)
18             Math.round(induk[0][k] + (r * x));
19         hasilmutasi[i][k] = hasilmutasitemp[0][k];
20     }

```

Kode Program 5.3 Implementasi proses mutasi

Berikut penjelasan Kode Program 5.3:

1. Baris 2-5, proses memilih induk yang akan dimutasi secara acak.
2. Baris 6-12, proses mencari nilai maksimum dan minimum.
3. Baris 15, proses menentukan nilai *r* secara acak.
4. Baris 16-20, proses *random mutation*.

5.2.4 Implementasi perhitungan penalti

Sebelum menghitung nilai *fitness* maka dihitung terlebih dahulu penalti masing-masing gizi, variasi makanan dan total harga setiap individu. Perhitungan penalti gizi terdiri dari penalti kalori, penalti karbohidrat, penalti protein, penalti lemak dan penalti natrium. Implementasi proses perhitungan penalti ditunjukkan pada Kode Program 5.4.

```

1  for (int i = 0; i < populasi_all.length; i++) {
2      for (int j = 0; j < tk[0].length; j++) {
3          if (tk[i][j] < kkalori) {
4              pk[i][j] = kkalori - tk[i][j];
5          } else if (tk[i][j] > kkalori) {

```

```

6         pk[i][j] = tk[i][j] - kkalori;
7     } else {
8         pk[i][j] = 0;
9     }
10 }
11 }
12 for (int i = 0; i < populasi_all.length; i++) {
13     for (int j = 0; j < tkb[0].length; j++) {
14         if (tkb[i][j] < kkarbohidrat) {
15             pkb[i][j] = kkarbohidrat - tkb[i][j];
16         } else if (tkb[i][j] > kkarbohidrat) {
17             pkb[i][j] = tkb[i][j] - kkarbohidrat;
18         } else {
19             pkb[i][j] = 0;
20         }
21     }
22 }
23 for (int i = 0; i < populasi_all.length; i++) {
24     for (int j = 0; j < tp[0].length; j++) {
25         if (tp[i][j] < kprotein) {
26             pp[i][j] = kprotein - tp[i][j];
27         } else if (tp[i][j] > kprotein) {
28             pp[i][j] = tp[i][j] - kprotein;
29         } else {
30             pp[i][j] = 0;
31         }
32     }
33 }
34 for (int i = 0; i < populasi_all.length; i++) {
35     for (int j = 0; j < tl[0].length; j++) {
36         if (tl[i][j] < klemak) {
37             pl[i][j] = klemak - tl[i][j];
38         } else if (tl[i][j] > klemak) {
39             pl[i][j] = tl[i][j] - klemak;
40         } else {
41             pl[i][j] = 0;
42         }
43     }
44 }
45 for (int i = 0; i < populasi_all.length; i++) {
46     for (int j = 0; j < tn[0].length; j++) {
47         if (tn[i][j] < 1000) {
48             pn[i][j] = 1000 - tn[i][j];
49         } else if (tn[i][j] > 1200) {
50             pn[i][j] = tn[i][j] - 1200;
51         } else {
52             pn[i][j] = 0;
53         }
54     }
55 }
56 double tpk = 0;
57 for (int i = 0; i < populasi_all.length; i++) {
58     for (int j = 0; j < pk[0].length; j++) {
59         tpk += pk[i][j];
60     }
61     totalpenalti[i][0] = tpk;
62     tpk = 0;
63 }
64 double tpkb = 0;

```

```

65 for (int i = 0; i < populasi_all.length; i++) {
66     for (int j = 0; j < pkb[0].length; j++) {
67         tpkb += pkb[i][j];
68     }
69     totalpenalti[i][1] = tpkb;
70     tpkb = 0;
71 }
72 double tpp = 0;
73 for (int i = 0; i < populasi_all.length; i++) {
74     for (int j = 0; j < pp[0].length; j++) {
75         tpp += pp[i][j];
76     }
77     totalpenalti[i][2] = tpp;
78     tpp = 0;
79 }
80 double tpl = 0;
81 for (int i = 0; i < populasi_all.length; i++) {
82     for (int j = 0; j < pl[0].length; j++) {
83         tpl += pl[i][j];
84     }
85     totalpenalti[i][3] = tpl;
86     tpl = 0;
87 }
88 double tpn = 0;
89 for (int i = 0; i < populasi_all.length; i++) {
90     for (int j = 0; j < pn[0].length; j++) {
91         tpn += pn[i][j];
92     }
93     totalpenalti[i][4] = tpn;
94     tpn = 0;
95 }
96 double[] totalharga = new
double[populasi_all.length];
97 for (int i = 0; i < populasi_all.length; i++) {
98     for (int j = 0; j < tharga[0].length; j++) {
99         totalharga[i] += tharga[i][j];
100     }
101 }
102 int[][] variasi = new
int[populasi_all.length][panjangkromosom];
103 int[] totalvariasi = new int[populasi_all.length];
104 for (int i = 0; i < populasi_all.length; i++) {
105     for (int j = 0; j < populasi_all[0].length;
        j++) {
106         variasi[i][j] = 1;
107     }
108 }
109 for (int i = 0; i < populasi_all.length; i++) {
110     for (int j = 0; j < populasi_all[0].length -
        10; j++) {
111         if (j == 4) {
112             j = 15;
113         } else if (j == 19) {
114             j = 30;
115         } else if (j == 34) {
116             j = 45;
117         } else if (j == 64) {
118             j = 75;
119         } else if (j == 79) {

```



```

120         j = 90;
121     } else if (j == 94) {
122         j = 105;
123     }
124     if (populasi_all[i][j] == populasi_all[i][j + 5]) {
125         variasi[i][j] = 0;
126         variasi[i][j + 5] = 0;
127     }
128     if (populasi_all[i][j] == populasi_all[i][j + 10]) {
129         variasi[i][j] = 0;
130         variasi[i][j + 10] = 0;
131     }
132     if (populasi_all[i][j+5] == populasi_all[i][j + 10])
133     {
134         variasi[i][j+5] = 0;
135         variasi[i][j + 10] = 0;
136     }
137 }
138 for (int i = 0; i < variasi.length; i++) {
139     for (int j = 0; j < variasi[0].length; j++) {
140         totalvariasi[i] += variasi[i][j];
141     }
142 }

```

Kode Program 5.4 Implementasi proses perhitungan penalti

Berikut penjelasan Kode Program 5.4:

1. Baris 1-10, proses menghitung penalti kalori.
2. Baris 12-22, proses menghitung penalti karbohidrat.
3. Baris 23-33, proses menghitung penalti protein.
4. Baris 34-44, proses menghitung penalti lemak.
5. Baris 45-55, proses menghitung penalti natrium.
6. Baris 56-63, proses menghitung total penalti kalori.
7. Baris 64-71, proses menghitung total penalti karbohidrat.
8. Baris 72-79, proses menghitung total penalti protein.
9. Baris 80-87, proses menghitung total penalti lemak.
10. Baris 88-95, proses menghitung total penalti natrium.
11. Baris 96-101, proses menghitung total harga.
12. Baris 102-139, proses menghitung total variasi.

5.2.5 Implementasi proses perhitungan *fitness*

Proses perhitungan *fitness* dilakukan untuk menentukan kualitas setiap individu. Nilai *fitness* diperoleh dengan cara menjumlahkan hasil bagi konstanta dengan penalti dan konstanta dengan total harga kemudian dijumlahkan dengan hasil kali konstanta dengan total variasi. Implementasi proses perhitungan *fitness* ditunjukkan pada Kode Program 5.5.

```

1  for (int i = 0; i < totalpenalti.length; i++) {
2      for (int j = 0; j < 7; j++) {
3          if (j == 0) {
4              nfitness += 1 / (1 + totalpenalti[i][j]);
5          }
6          if (j == 1) {
7              nfitness += 1 / (1 + totalpenalti[i][j]);
8          }
9          if (j == 2) {
10             nfitness += 50 / (1 + totalpenalti[i][j]);
11         }
12         if (j == 3) {
13             nfitness += 20 / (1 + totalpenalti[i][j]);
14         }
15         if (j == 4) {
16             nfitness += 50 / (1 + totalpenalti[i][j]);
17         }
18         if (j == 5) {
19             nfitness += 100 / totalharga[i];
20         }
21         if (j == 6) {
22             nfitness += 5 * totalvariasi[i];
23         }
24     }
25     fitness[i] = nfitness;
26     nfitness = 0;
27 }

```

Kode Program 5.5 Implementasi proses perhitungan *fitness*

Berikut penjelasan Kode Program 5.5:

1. Baris 1, perulangan untuk menghitung *fitness* berdasarkan setiap total penalti masing-masing individu.
2. Baris 2-27, proses menghitung *fitness* berdasarkan total penalti masing-masing kandungan gizi, total harga dan variasi.

5.2.6 Implementasi proses seleksi

Proses seleksi dilakukan untuk menentukan individu-individu yang dipertahankan pada generasi selanjutnya. Seluruh individu diurutkan berdasarkan nilai *fitness* dari tertinggi hingga terendah kemudian mengambil individu dengan nilai *fitness* tertinggi sejumlah *popsiz*. Implementasi proses seleksi ditunjukkan pada Kode Program 5.6.

```

1  for (int i = 0; i < fitness.length - 1; i++) {
2      for (int j = 0; j < fitness.length - i - 1; j++) {
3          if (fitness[j] < fitness[j + 1]) {
4              temp = fitness[j];
5              fitness[j] = fitness[j + 1];
6              fitness[j + 1] = temp;
7              for (int k = 0; k < panjangkromosom; k++) {
8                  int temp2 = populasi_all[j][k];
9                  populasi_all[j][k] = populasi_all[j + 1][k];
10                 populasi_all[j + 1][k] = temp2;
11             }
12         }
13     }

```

```

14 }
15 for (int i = 0; i < popsize; i++) {
16     for (int j = 0; j < panjangkromosom; j++) {
17         populasi[i][j] = populasi_all[i][j];
18     }
19 }

```

Kode Program 5.6 Implementasi proses seleksi

Berikut penjelasan Kode Program 5.6:

1. Baris 1-14, proses mengurutkan individu berdasarkan nilai *fitness* tertinggi hingga terendah.
2. Baris 15-19, proses mengambil individu yang memiliki nilai *fitness* tertinggi sebanyak *popsize*.

5.3 Implementasi antarmuka

Implementasi antarmuka pada sistem ini terdiri dari beberapa halaman yaitu halaman input, halaman proses algoritme genetika, halaman hasil rekomendasi menu makanan dan halaman daftar makanan.

5.3.1 Implementasi antarmuka halaman utama

Pada implementasi antarmuka halaman utama terdapat kolom yang disediakan untuk pengguna melakukan input data ibu hamil yang terdiri dari nama, usia ibu hamil, usia kehamilan, tinggi badan, berat badan, tekanan darah dan aktivitas serta terdapat kolom untuk input parameter algoritme genetika dan jumlah hari makanan yang diinginkan. Pada halaman ini juga ditampilkan hasil perhitungan gizi berdasarkan data input. Implementasi antarmuka halaman utama ditunjukkan pada Gambar 5.1.

SISTEM OPTIMASI ASUPAN MAKANAN HARIAN IBU HAMIL PENDERITA HIPERTENSI

Data Input

Nama: Y

Usia: 20

Usia Kehamilan: 19 minggu

Berat Badan: 53 kg

Tinggi Badan: 156 cm

Tekanan Darah: 160/90 mmHg

Aktivitas: Sedang

Jumlah Hari: 7 Hari

Operator Algoritme Genetika

Generasi: 240

PopSize: 90

Crossover Rate: 0.6

Mutation Rate: 0.5

Default

Submit

Kebutuhan Gizi Harian

Usia	Usia Kehamilan	Berat Badan	Tinggi Badan	Tekanan Darah	Aktivitas
20	19	53	156	160/90	Sedang

Kebutuhan Gizi Harian

Kebutuhan Kalori	1736.0	kcal
Kebutuhan Karbohidrat	322.1	gram
Kebutuhan Protein	85.1	gram
Kebutuhan Lemak	48.578	gram
Kebutuhan Natrium	1000-1200	mg

Gambar 5.1 Antarmuka halaman utama

5.3.2 Implementasi antarmuka halaman proses algoritme genetika

5.3.2.1 Implementasi antarmuka halaman populasi awal

Pada halaman populasi awal akan menampilkan kromosom populasi awal di setiap generasi. Halaman populasi awal ditunjukkan pada Gambar 5.2.

Gambar 5.2 Antarmuka halaman populasi awal

5.3.2.2 Implementasi antarmuka halaman hasil crossover

Pada halaman hasil *crossover* akan menampilkan populasi hasil *crossover* pada setiap generasi. Halaman hasil *crossover* dapat dilihat pada Gambar 5.3.

Gambar 5.3 Antarmuka halaman hasil crossover

5.3.2.3 Implementasi antarmuka halaman hasil mutasi

Pada halaman hasil mutasi akan menampilkan populasi hasil mutasi pada setiap generasi. Halaman hasil mutasi dapat dilihat pada Gambar 5.4.

Generasi ke-	Individu	Kromosom
Generasi ke-1	C1	21 33 24 13 29 15 30 21 31 19 19 34 15 30 27 16 25 18 32 26 27 13 13 23 15 34 20 14 31 31 23 25 23 33 17 2...
Generasi ke-1	C2	41 52 42 28 42 30 50 35 29 40 39 37 42 38 36 47 50 34 42 27 44 51 37 46 48 29 39 30 41 35 30 47 32 27 27 4...
Generasi ke-1	C3	25 17 25 32 38 26 31 19 38 22 30 23 26 38 37 30 36 25 32 22 32 30 31 31 35 27 21 28 35 23 19 25 25 31 2...
Generasi ke-1	C4	32 22 20 31 29 35 25 30 20 18 16 35 17 24 19 36 26 23 19 19 34 26 19 21 21 26 34 23 38 23 30 18 16 20 31 2...
Generasi ke-1	C5	33 36 27 30 36 42 28 31 33 34 35 35 38 39 49 43 40 33 45 40 34 32 31 20 37 28 32 31 41 46 36 32 29 43 41 3...
Generasi ke-1	C6	36 35 29 23 37 39 25 24 43 22 33 45 34 40 36 33 28 28 32 26 23 41 25 22 24 33 24 23 44 29 30 25 24 24 44 4...
Generasi ke-1	C7	26 27 30 33 41 42 37 21 36 29 22 26 25 37 43 41 20 37 28 36 32 37 33 23 21 31 43 22 35 43 23 22 30 34 27 2...
Generasi ke-1	C8	38 23 28 26 36 37 33 35 23 33 30 29 34 40 19 30 30 24 35 23 25 25 20 23 39 37 31 29 24 26 43 32 26 38 2...
Generasi ke-1	C9	16 27 20 20 28 5 5 22 22 15 9 29 12 12 13 19 11 20 16 26 7 24 22 20 20 29 28 18 25 26 13 14 19 6 8 10 25 7 1...
Generasi ke-1	C10	29 19 13 20 22 16 12 21 24 16 38 31 21 17 17 13 18 25 23 12 25 34 14 20 12 16 24 28 35 19 24 33 28 13 31 2...
Generasi ke-1	C11	18 34 12 33 17 19 26 15 31 28 23 20 15 27 25 24 31 13 14 25 24 26 13 29 22 15 37 16 25 24 20 12 26 20 27 1...
Generasi ke-1	C12	43 27 31 35 42 38 24 30 34 41 34 27 26 33 23 29 31 25 34 45 40 25 30 45 25 47 25 28 29 24 38 29 44 46 2...
Generasi ke-1	C13	23 28 22 21 26 28 24 17 33 16 16 37 25 18 39 22 17 28 17 26 21 21 17 38 32 19 35 20 31 20 36 27 31 26 31 2...
Generasi ke-1	C14	12 13 16 19 27 28 23 7 22 15 8 12 11 23 29 27 6 23 14 22 18 23 19 19 7 17 29 8 21 29 9 8 16 20 13 13 19 18...
Generasi ke-1	C15	44 29 34 32 42 43 39 41 29 39 36 36 35 40 45 25 36 36 30 41 29 31 31 26 29 45 43 37 35 30 32 49 38 32 44 3...
Generasi ke-1	C16	27 38 31 36 38 16 16 33 33 26 20 40 23 23 24 30 22 31 27 37 18 35 33 31 31 20 38 29 36 37 24 25 30 17 29 2...
Generasi ke-1	C17	23 19 21 38 31 20 18 20 20 22 38 33 25 19 39 34 22 20 34 31 18 33 32 35 19 16 41 26 16 25 33 29 33 20 16 1...
Generasi ke-1	C18	20 16 18 35 28 17 15 17 17 19 35 30 23 16 38 31 19 17 31 28 15 30 29 32 16 13 38 23 13 22 30 26 30 17 13 1...
Generasi ke-1	C19	36 44 34 26 29 35 44 29 40 23 45 43 38 34 43 24 38 26 39 23 28 36 23 39 45 24 44 28 34 40 26 39 39 33 42 2...
Generasi ke-1	C20	37 23 23 34 29 36 30 20 16 37 26 26 25 36 31 28 25 23 33 21 33 25 25 23 22 24 22 19 38 30 24 32 20 16 18 2...
Generasi ke-1	C21	29 38 36 42 33 27 30 30 40 25 30 20 25 31 42 33 28 22 23 26 31 26 28 37 27 21 42 23 26 20 21 31 31 19 25 3...
Generasi ke-1	C22	16 33 14 25 19 10 32 21 29 19 12 28 12 18 26 11 15 17 10 33 19 29 15 33 13 26 32 24 16 13 23 20 18 16 28 3...
Generasi ke-1	C23	37 41 35 29 41 43 40 34 26 25 45 50 39 44 25 46 30 42 42 29 45 48 41 48 41 27 38 40 46 30 33 37 30 28 47 3...
Generasi ke-1	C24	29 27 14 22 24 27 17 21 16 10 16 27 13 15 14 16 12 11 12 28 20 8 12 30 26 18 11 18 11 8 17 22 16 9 11 25 1...
Generasi ke-1	C25	46 30 33 28 23 26 27 32 36 30 37 36 40 25 45 43 31 41 45 38 24 31 27 39 38 31 25 28 42 30 46 47 34 31 44 3...
Generasi ke-1	C26	16 12 14 31 24 13 11 13 15 31 26 19 12 32 27 15 13 27 24 11 26 25 28 12 9 34 19 9 18 26 22 26 13 12 12 2...
Generasi ke-1	C27	29 29 37 45 46 29 37 27 44 41 35 44 40 35 23 27 31 30 23 27 33 26 39 31 26 34 29 35 33 23 37 36 33 46 2...
Generasi ke-1	C28	26 34 22 20 22 32 24 25 20 27 27 29 24 28 30 27 37 25 33 41 21 25 28 31 19 34 40 31 38 40 24 34 33 26 20 2...
Generasi ke-1	C29	14 17 8 12 9 17 9 18 9 8 24 23 5 19 24 6 22 7 16 5 22 18 4 16 19 25 19 11 7 10 18 17 14 7 25 9 5 16 16 15 20 2...
Generasi ke-1	C30	27 28 31 34 42 43 38 22 37 30 23 27 28 38 44 42 21 38 29 37 33 38 34 34 22 32 44 23 36 44 24 23 31 35 28 2...

Gambar 5.4 Antarmuka halaman hasil mutasi

5.3.2.4 Implementasi antarmuka halaman individu terbaik

Pada halaman individu terbaik akan menampilkan individu-individu terbaik pada setiap generasi beserta nilai *fitness*. Halaman individu terbaik dapat dilihat pada Gambar 5.5.

Generasi ke-	Kromosom	Fitness
Generasi ke-1	18 16 10 5 11 20 24 5 3 6 9 5 1 19 15 20 16 5 23 1 22 1 10 13 23 19 14 8 14 5 12 9 8 9 4 10 19 5 8 12 9 7 1 ...	525.0427917072813
Generasi ke-2	18 16 10 5 11 20 24 5 3 6 9 5 1 19 15 20 16 5 23 1 22 1 10 13 23 19 14 8 14 5 12 9 8 9 4 10 19 5 8 12 9 7 1 ...	525.0427917072813
Generasi ke-3	8 16 8 14 4 14 7 4 4 7 17 9 6 4 14 10 23 11 17 14 7 17 13 14 15 11 21 9 6 19 8 19 6 12 16 14 9 21 9 10 2 13 ...	525.0587696046352
Generasi ke-4	8 16 8 14 4 14 7 4 4 7 17 9 6 4 14 10 23 11 17 14 7 17 13 14 15 11 21 9 6 19 8 19 6 12 16 14 9 21 9 10 2 13 ...	525.0587696046352
Generasi ke-5	8 16 8 14 4 14 7 4 4 7 17 9 6 4 14 10 23 11 17 14 7 17 13 14 15 11 21 9 6 19 8 19 6 12 16 14 9 21 9 10 2 13 ...	525.0587696046352
Generasi ke-6	8 16 8 14 4 14 7 4 4 7 17 9 6 4 14 10 23 11 17 14 7 17 13 14 15 11 21 9 6 19 8 19 6 12 16 14 9 21 9 10 2 13 ...	525.0587696046352
Generasi ke-7	8 16 8 14 4 14 7 4 4 7 17 9 6 4 14 10 23 11 17 14 7 17 13 14 15 11 21 9 6 19 8 19 6 12 16 14 9 21 9 10 2 13 ...	525.0587696046352
Generasi ke-8	8 16 8 14 4 14 7 4 4 7 17 9 6 4 14 10 23 11 17 14 7 17 13 14 15 11 21 9 6 19 8 19 6 12 16 14 9 21 9 10 2 13 ...	525.0587696046352
Generasi ke-9	8 16 8 14 4 14 7 4 4 7 17 9 6 4 14 10 23 11 17 14 7 17 13 14 15 11 21 9 6 19 8 19 6 12 16 14 9 21 9 10 2 13 ...	525.0587696046352
Generasi ke-10	8 16 8 14 4 14 7 4 4 7 17 9 6 4 14 10 23 11 17 14 7 17 13 14 15 11 21 9 6 19 8 19 6 12 16 14 9 21 9 10 2 13 ...	525.0587696046352
Generasi ke-11	8 16 8 14 4 14 7 4 4 7 17 9 6 4 14 10 23 11 17 14 7 17 13 14 15 11 21 9 6 19 8 19 6 12 16 14 9 21 9 10 2 13 ...	525.0587696046352
Generasi ke-12	8 16 8 14 4 14 7 4 4 7 17 9 6 4 14 10 23 11 17 14 7 17 13 14 15 11 21 9 6 19 8 19 6 12 16 14 9 21 9 10 2 13 ...	525.0587696046352
Generasi ke-13	8 16 8 14 4 14 7 4 4 7 17 9 6 4 14 10 23 11 17 14 7 17 13 14 15 11 21 9 6 19 8 19 6 12 16 14 9 21 9 10 2 13 ...	525.0587696046352
Generasi ke-14	8 16 8 14 4 14 7 4 4 7 17 9 6 4 14 10 23 11 17 14 7 17 13 14 15 11 21 9 6 19 8 19 6 12 16 14 9 21 9 10 2 13 ...	525.0587696046352
Generasi ke-15	8 16 8 14 4 14 7 4 4 7 17 9 6 4 14 10 23 11 17 14 7 17 13 14 15 11 21 9 6 19 8 19 6 12 16 14 9 21 9 10 2 13 ...	525.0587696046352
Generasi ke-16	8 16 8 14 4 14 7 4 4 7 17 9 6 4 14 10 23 11 17 14 7 17 13 14 15 11 21 9 6 19 8 19 6 12 16 14 9 21 9 10 2 13 ...	525.0587696046352
Generasi ke-17	8 16 8 14 4 14 7 4 4 7 17 9 6 4 14 10 23 11 17 14 7 17 13 14 15 11 21 9 6 19 8 19 6 12 16 14 9 21 9 10 2 13 ...	525.0587696046352
Generasi ke-18	8 16 8 14 4 14 7 4 4 7 17 9 6 4 14 10 23 11 17 14 7 17 13 14 15 11 21 9 6 19 8 19 6 12 16 14 9 21 9 10 2 13 ...	525.0587696046352
Generasi ke-19	8 16 8 14 4 14 7 4 4 7 17 9 6 4 14 10 23 11 17 14 7 17 13 14 15 11 21 9 6 19 8 19 6 12 16 14 9 21 9 10 2 13 ...	525.0587696046352
Generasi ke-20	8 16 8 14 4 14 7 4 4 7 17 9 6 4 14 10 23 11 17 14 7 17 13 14 15 11 21 9 6 19 8 19 6 12 16 14 9 21 9 10 2 13 ...	525.0587696046352
Generasi ke-21	8 16 8 14 4 14 7 4 4 7 17 9 6 4 14 10 23 11 17 14 7 17 13 14 15 11 21 9 6 19 8 19 6 12 16 14 9 21 9 10 2 13 ...	525.0587696046352
Generasi ke-22	8 16 8 14 4 14 7 4 4 7 17 9 6 4 14 10 23 11 17 14 7 17 13 14 15 11 21 9 6 19 8 19 6 12 16 14 9 21 9 10 2 13 ...	525.0587696046352
Generasi ke-23	8 16 8 14 4 14 7 4 4 7 17 9 6 4 14 10 23 11 17 14 7 17 13 14 15 11 21 9 6 19 8 19 6 12 16 14 9 21 9 10 2 13 ...	525.0587696046352
Generasi ke-24	8 16 8 14 4 14 7 4 4 7 17 9 6 4 14 10 23 11 17 14 7 17 13 14 15 11 21 9 6 19 8 19 6 12 16 14 9 21 9 10 2 13 ...	525.0587696046352
Generasi ke-25	8 16 8 14 4 14 7 4 4 7 17 9 6 4 14 10 23 11 17 14 7 17 13 14 15 11 21 9 6 19 8 19 6 12 16 14 9 21 9 10 2 13 ...	525.0587696046352
Generasi ke-26	8 16 8 14 4 14 7 4 4 7 17 9 6 4 14 10 23 11 17 14 7 17 13 14 15 11 21 9 6 19 8 19 6 12 16 14 9 21 9 10 2 13 ...	525.0587696046352
Generasi ke-27	8 16 8 14 4 14 7 4 4 7 17 9 6 4 14 10 23 11 17 14 7 17 13 14 15 11 21 9 6 19 8 19 6 12 16 14 9 21 9 10 2 13 ...	525.0587696046352
Generasi ke-28	8 16 8 14 4 14 7 4 4 7 17 9 6 4 14 10 23 11 17 14 7 17 13 14 15 11 21 9 6 19 8 19 6 12 16 14 9 21 9 10 2 13 ...	525.0587696046352
Generasi ke-29	8 16 8 14 4 14 7 4 4 7 17 9 6 4 14 10 23 11 17 14 7 17 13 14 15 11 21 9 6 19 8 19 6 12 16 14 9 21 9 10 2 13 ...	525.0587696046352
Generasi ke-30	8 16 8 14 4 14 7 4 4 7 17 9 6 4 14 10 23 11 17 14 7 17 13 14 15 11 21 9 6 19 8 19 6 12 16 14 9 21 9 10 2 13 ...	525.0587696046352

Gambar 5.5 Antarmuka halaman individu terbaik

5.3.3 Implementasi antarmuka halaman hasil menu makanan

Pada halaman hasil menu makanan akan menampilkan menu makanan yang direkomendasikan oleh sistem sejumlah hari yang telah ditentukan. Halaman hasil menu makanan dapat dilihat pada Gambar 5.6.

Gambar 5.6 Antarmuka halaman hasil menu makanan

5.3.4 Implementasi antarmuka halaman daftar makanan

Pada halaman daftar makanan akan menampilkan daftar makanan yang digunakan sistem dalam memberikan rekomendasi menu makanan oleh sistem beserta kadar kandungan gizi dan harga. Halaman daftar makanan dapat dilihat pada Gambar 5.7.

Gambar 5.7 Antarmuka halaman daftar makanan

BAB 6 PENGUJIAN DAN PEMBAHASAN

Bab ini akan menjelaskan tentang pengujian yang dilakukan dan hasil yang didapatkan berdasarkan skenario pengujian yang digunakan. Pengujian tersebut terdiri dari pengujian ukuran generasi, pengujian ukuran populasi, dan pengujian kombinasi nilai cr , mr .

6.1 Pengujian ukuran generasi

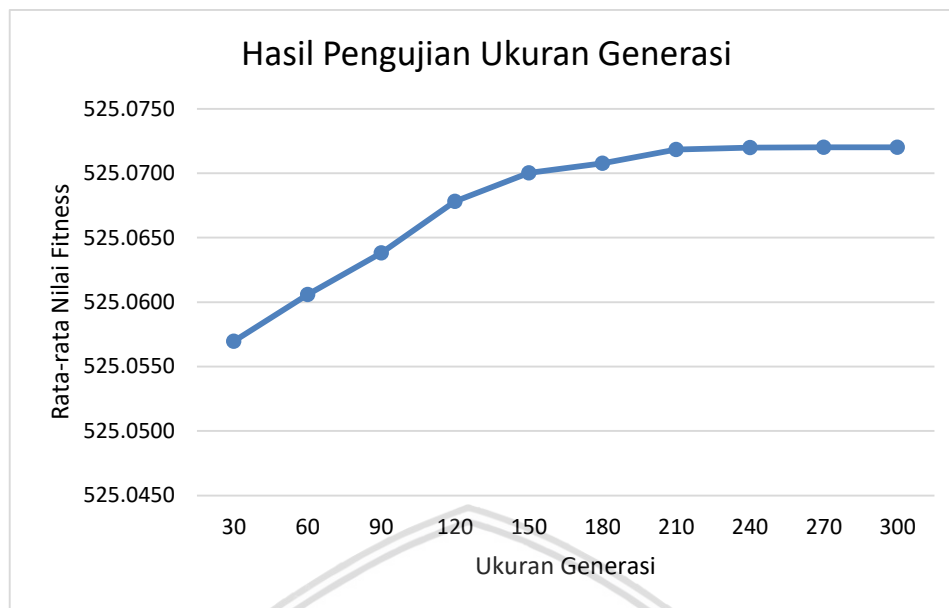
Pengujian ukuran generasi yang dilakukan yaitu ukuran generasi 30 hingga 300. Pengujian ini dilakukan untuk mengetahui berapa ukuran generasi optimal yang dapat digunakan dan bagaimana pengaruh jumlah generasi terhadap rata-rata nilai *fitness*. Ukuran generasi yang diuji adalah 30, 60, 90, 120, 150, 180, 210, 240, 270, dan 300. Setiap pengujian ukuran generasi dilakukan sebanyak 10 kali uji coba. Ukuran populasi yang digunakan yaitu 70 dengan kombinasi nilai cr dan mr adalah 0.6 dan 0.4. Hasil pengujian jumlah generasi ditunjukkan pada Tabel 6.1 untuk hasil lengkapnya dapat dilihat pada Lampiran C.

Tabel 6.1 Hasil pengujian ukuran generasi

Generasi	Pengujian ke-					Rata-rata Nilai <i>Fitness</i>
	1	2	9	10	
30	525.0591	525.0536	525.0654	525.0516	525.0570
60	525.0678	525.0630	525.0654	525.0597	525.0606
90	525.0732	525.0630	525.0685	525.0676	525.0638
120	525.0733	525.0630	525.0784	525.0710	525.0678
150	525.0731	525.0630	525.0780	525.0710	525.0700
180	525.0733	525.0630	525.0798	525.0710	525.0708
210	525.0733	525.0647	525.0887	525.0710	525.0718
240	525.0730	525.0652	525.0898	525.0710	525.0720
270	525.0734	525.0653	525.0898	525.0710	525.0720
300	525.0734	525.0653	525.0898	525.0710	525.0720

Pada Tabel 6.1 hasil pengujian ukuran generasi dapat dilihat rata-rata nilai *fitness* tertinggi yang dihasilkan adalah 525,0720 yaitu pada ukuran generasi 240 dan rata-rata nilai *fitness* terendah adalah 525,0570 yaitu pada ukuran generasi 30 hal ini disebabkan ukuran generasi yang kecil membatasi eksplorasi ruang pencarian sebaliknya ukuran generasi yang besar memberi peluang untuk melakukan eksplorasi ruang pencarian yang lebih besar.

Berdasarkan hasil pengujian pada Tabel 6.1 dapat dilihat pengaruh ukuran generasi terhadap rata-rata nilai *fitness* yang didapatkan seperti ditunjukkan pada Gambar 6.1.



Gambar 6.1 Grafik hasil pengujian ukuran generasi

Pada Gambar 6.1 menunjukkan grafik rata-rata nilai *fitness* mengalami kenaikan pada generasi 30 hingga generasi 240 dan setelah generasi 240 rata-rata nilai *fitness* tidak mengalami perubahan yang signifikan, hal ini menunjukkan bahwa nilai *fitness* pada generasi tersebut telah mengalami konvergensi. Apabila pengujian ukuran generasi dilanjutkan dengan ukuran generasi lebih tinggi maka nilai *fitness* tidak memiliki selisih yang besar dan individu yang dihasilkan tidak berbeda jauh dengan induknya (Shafaat, 2017). Berdasarkan hasil tersebut maka ukuran generasi 240 dapat dianggap optimal untuk digunakan dalam optimasi asupan makanan harian ibu hamil penderita Hipertensi.

6.2 Pengujian ukuran populasi

Pada pengujian ukuran populasi dilakukan uji coba terhadap ukuran populasi kelipatan 10 dengan masing-masing 10 kali uji coba. Ukuran populasi yang diuji yaitu ukuran populasi 10 hingga 110. Hasil Pengujian ukuran populasi dapat dilihat pada Tabel 6.2 untuk hasil lengkapnya dapat dilihat pada Lampiran C.

Tabel 6.2 Hasil pengujian ukuran populasi

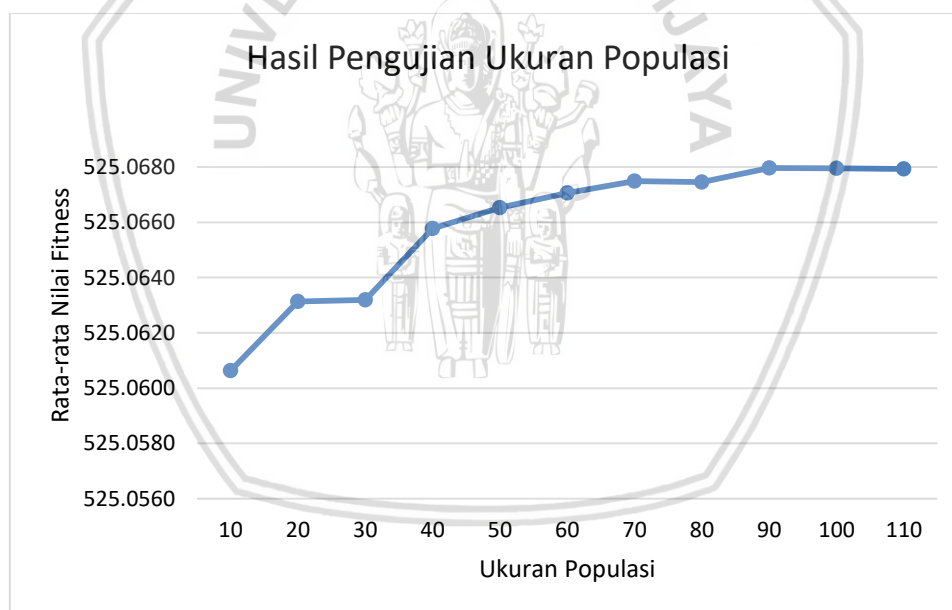
Populasi	Pengujian Ke-					Rata-rata Nilai <i>Fitness</i>
	1	2	9	10	
10	525.0663	525.0619	525.0539	525.0606	525.0606
20	525.0619	525.0528	525.0656	525.0540	525.0631
30	525.0589	525.0604	525.0608	525.0516	525.0632
40	525.0798	525.0589	525.0601	525.0625	525.0658
50	525.0687	525.0635	525.0646	525.0775	525.0665
60	525.0586	525.0638	525.0776	525.0841	525.0671
70	525.0653	525.0657	525.0650	525.0646	525.0675
80	525.0675	525.0695	525.0635	525.0648	525.0675

Tabel 6.2 Hasil pengujian ukuran populasi (lanjutan)

Populasi	Pengujian Ke-					Rata-rata Nilai <i>Fitness</i>
	1	2	9	10	
90	525.0612	525.0665	525.0613	525.0611	525.0680
100	525.0849	525.0605	525.0692	525.0631	525.0680
110	525.0596	525.0620	525.0759	525.0675	525.0679

Pada Tabel 6.2 hasil pengujian ukuran populasi dapat dilihat rata-rata nilai *fitness* tertinggi yang dihasilkan adalah 525,0680 yaitu pada ukuran populasi 90 dan rata-rata nilai *fitness* terendah adalah 525,0606 yaitu pada ukuran populasi 10, hal ini disebabkan ukuran populasi yang besar akan menghasilkan individu yang semakin beragam melalui proses *crossover* dan mutasi, dan berdampak pada nilai variasi yang dihasilkan, namun ukuran populasi yang besar juga tidak menjamin akan menghasilkan nilai *fitness* yang tinggi dikarenakan pembangkitan populasi yang dilakukan secara acak (Kusumaningsih, 2016). Ukuran populasi yang terlalu kecil akan memperkecil peluang untuk memperoleh solusi terbaik.

Berdasarkan hasil pengujian ukuran populasi pada Tabel 6.2 dapat dilihat grafik hasil pengujian seperti ditunjukkan pada Gambar 6.2.



Gambar 6.2 Grafik hasil pengujian ukuran populasi

Pada Gambar 6.2 dapat dilihat rata-rata nilai *fitness* mengalami kenaikan pada ukuran populasi 10 hingga 90. Pada ukuran populasi 100 rata-rata nilai *fitness* tidak menunjukkan perubahan yang signifikan hal ini menunjukkan bahwa telah mengalami konvergensi. Ukuran populasi 90 dapat dianggap sebagai ukuran populasi yang optimal.

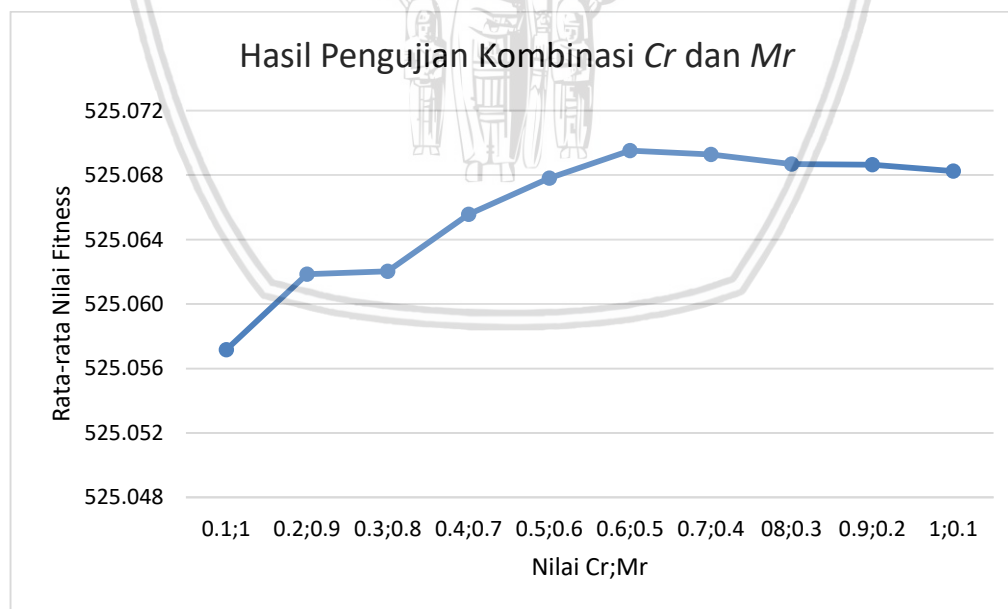
6.3 Pengujian kombinasi *crossover rate* dan *mutation rate*

Pada pengujian kombinasi *crossover rate* (*cr*) dan *mutation rate* (*mr*) dilakukan uji coba dengan kombinasi nilai *cr* dan *mr* antara 0.1 hingga 1. Uji coba setiap kombinasi *cr* dan *mr* dilakukan sebanyak 10 kali. Hasil Pengujian kombinasi *cr* dan *mr* dapat dilihat pada Tabel 6.3 untuk hasil lengkapnya dapat dilihat pada Lampiran C.

Tabel 6.3 Hasil pengujian kombinasi *cr* dan *mr*

<i>Cr; Mr</i>	Pengujian Ke-					Rata-rata Nilai <i>Fitness</i>
	1	2	9	10	
0.1 ; 1	525.0601	525.0511	525.0556	525.0572	525.0572
0.2 ; 0.9	525.0725	525.0622	525.0605	525.0596	525.0619
0.3 ; 0.8	525.0581	525.0601	525.062	525.0658	525.062
0.4 ; 0.7	525.0538	525.0711	525.0836	525.0703	525.0656
0.5 ; 0.6	525.0695	525.0636	525.0637	525.0767	525.0678
0.6 ; 0.5	525.0665	525.058	525.0764	525.0682	525.0695
0.7 ; 0.4	525.0712	525.082	525.0613	525.0732	525.0693
0.8 ; 0.3	525.0688	525.0766	525.0655	525.0649	525.0687
0.9 ; 0.2	525.0961	525.0621	525.0636	525.0772	525.0687
1 ; 0.1	525.0594	525.0597	525.0657	525.0614	525.0682

Berdasarkan hasil pengujian pada Tabel 6.3 dapat dilihat grafik hasil pengujian seperti ditunjukkan pada Gambar 6.2.



Gambar 6.3 Grafik hasil pengujian kombinasi *cr* dan *mr*

Pada Tabel 6.3 dapat dilihat rata-rata nilai *fitness* tertinggi didapatkan pada kombinasi nilai *cr*=0,6 dan *mr*=0,5 yaitu sebesar 525,0695 dan rata-rata nilai *fitness* terendah ketika kombinasi nilai *cr*=0,1 dan *mr*=1 yaitu sebesar 525,0572. Gambar 6.3 menunjukkan grafik peningkatan rata-rata nilai *fitness* pada kombinasi nilai

$cr=0.1$ dan $mr=1$ sampai dengan kombinasi nilai $cr=0.6$ dan $mr=0.5$. Berdasarkan hasil tersebut menunjukkan apabila nilai *crossover rate* yang terlalu rendah dan *mutation rate* yang terlalu tinggi menyebabkan Algoritme genetika memiliki kemampuan eksplorasi yang lebih besar sehingga individu yang dihasilkan beragam, dan sebaliknya nilai *crossover rate* yang terlalu tinggi dan *mutation rate* yang terlalu rendah menyebabkan algoritme genetika memiliki tingkat eksploitasi yang besar sehingga individu yang terbentuk cenderung mirip dengan induknya dengan nilai fitness yang cenderung menurun serta membatasi eksplorasi terhadap individu yang lainnya (Siahaan, 2017). Berdasarkan hasil pengujian kombinasi nilai cr dan mr optimal yang dapat digunakan adalah $cr=0,6$ dan $mr=0,5$.

6.4 Hasil analisis global

Berdasarkan hasil pengujian ukuran generasi, ukuran populasi dan kombinasi *crossover rate* dan *mutation rate* didapatkan nilai optimal parameter algoritme genetika sebagai berikut:

- Ukuran generasi : 240
- Ukuran populasi : 90
- *Crossover rate* : 0.6
- *Mutation rate* : 0.5

Parameter algoritme genetika hasil pengujian tersebut selanjutnya digunakan dalam pengujian terhadap 5 data ibu hamil dengan hasil kebutuhan gizi aktual seperti ditunjukkan pada Tabel 6.4.

Tabel 6.4 Hasil perhitungan kebutuhan gizi aktual

Nama	Kalori (kkal)	Karbohidrat (gr)	Protein (gr)	Lemak (gr)	Natrium (mg)
INM	1449	241,713	59,338	35,867	1000-1200
ZR	1449	241,713	59,338	35,867	1000-1200
IA	1665	276,813	67,438	37,667	1000-1200
YM	1692	281,200	68,450	38,267	1000-1200
ND	1938	354,925	92,675	53,067	1000-1200

Berdasarkan Tabel 6.4 dapat dihitung selisih kebutuhan gizi aktual dengan kandungan gizi hasil rekomendasi sistem yang dapat dilihat pada Tabel 6.5.

Tabel 6.5 Selisih kebutuhan gizi dan kandungan gizi hasil sistem

Nama	Kalori (kkal)	Karbohidrat (gr)	Protein (gr)	Lemak (gr)	Natrium (mg)
INM	115,11	23,02	3,58	2,55	-490,21
	(7,94%)	(9,52%)	(6,03%)	(7,12%)	(-49,02%)
ZR	91,06	20,24	5,63	2,91	-348,86
	(6,28%)	(8,37%)	(9,49%)	(8,13%)	(-34,89%)
IA	101,81	16,27	4,65	2,57	-444,71
	(6,11%)	(5,88%)	(6,89%)	(6,82%)	(-44,47%)

Tabel 6.5 Selisih kebutuhan gizi dan kandungan gizi hasil sistem (lanjutan)

Nama	Kalori (kkal)	Karbohidrat (gr)	Protein (gr)	Lemak (gr)	Natrium (mg)
YM	12,27	-2,21	-3,56	3,19	-484,41
	(0,73%)	(-0,79%)	(-5,21%)	(8,34%)	(-48,44%)
ND	-168,51	-18,63	-8,38	-2,47	-352,43
	(-8,70%)	(-5,25%)	(-9,05%)	(-4,65%)	(-35,24%)

Pada Tabel 6.5 dapat dilihat rata-rata selisih kandungan gizi makanan hasil rekomendasi sistem untuk 4 komponen kandungan gizi sesuai dengan batas toleransi pakar dari kebutuhan gizi aktual yang ditentukan, sehingga hasil rekomendasi sistem untuk kebutuhan kalori, karbohidrat, protein, dan lemak dapat memenuhi kebutuhan gizi ibu hamil penderita Hipertensi. Untuk kandungan gizi natrium hasil rekomendasi sistem kurang dari batas toleransi pakar. Hal ini disebabkan karena proses inisialisasi populasi awal yang secara acak sehingga kemungkinan individu yang dihasilkan masih jauh dari hasil solusi optimal, selain itu jumlah data makanan yang digunakan memiliki kandungan natrium cukup masih kurang beragam, sehingga pencarian solusi menjadi terbatas.



BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan mengenai optimasi asupan makanan harian ibu hamil penderita Hipertensi didapatkan kesimpulan sebagai berikut

1. Algoritme genetika dapat diimplementasikan dalam optimasi asupan makanan harian ibu hamil penderita Hipertensi dengan memberikan rekomendasi makanan untuk beberapa hari yang terdiri dari makan pagi, makan siang, dan makan malam. Representasi kromosom menggunakan representasi bilangan integer yang merepresentasikan id makanan, metode *crossover* yang digunakan adalah *extended intermediate crossover*, metode mutasi yang digunakan adalah *random mutation*, dan proses seleksi menggunakan seleksi elitism. Langkah-langkah dalam melakukan optimasi adalah sebagai berikut:
 - a. Melakukan perhitungan kebutuhan gizi berdasarkan parameter data ibu hamil.
 - b. Melakukan proses inisialisasi populasi awal berdasarkan parameter algoritme genetika yang ditentukan.
 - c. Melakukan proses *crossover* menggunakan metode *extended intermediate crossover*.
 - d. Melakukan proses mutasi menggunakan metode *random mutation*.
 - e. Melakukan perhitungan nilai *fitness* masing-masing individu untuk menentukan kualitas individu.
 - f. Melakukan proses seleksi untuk menentukan populasi pada generasi berikutnya dan solusi terbaik.
2. Kualitas hasil rekomendasi makanan sistem dapat diukur dengan menggunakan parameter algoritme genetika berdasarkan hasil pengujian antara lain ukuran generasi sebesar 240, ukuran populasi sebesar 90 dan kombinasi nilai *cr* dan *mr* adalah 0.6 dan 0.5.

7.2 Saran

Untuk pengembangan pada penelitian selanjutnya terdapat beberapa saran yang diharapkan dapat memperbaiki hasil penelitian yaitu sebagai berikut:

1. Data yang digunakan pada penelitian selanjutnya diharapkan lebih beragam terutama memperhatikan pada kandungan natriumnya, agar solusi yang diberikan lebih baik dan sesuai dengan kebutuhan natrium harian.
2. Pada penelitian selanjutnya diharapkan dapat memodifikasi perhitungan nilai *fitness* terutama pada penentuan konstanta prioritas untuk

kebutuhan gizi yaitu natrium, agar dapat memberikan hasil yang lebih baik dan solusi yang diberikan sesuai dengan kebutuhan gizi dan batas toleransi pakar.

3. Pada penelitian selanjutnya diharapkan dapat mempertimbangkan biaya makanan untuk mendapatkan hasil rekomendasi makanan dengan harga yang terjangkau.



DAFTAR PUSTAKA

- Almatsier, S., 2001. *Prinsip Dasar Ilmu Gizi*. s.l.:Gramedia Pustaka Utama.
- Ariani, A. P., 2017. *Ilmu Gizi*. Yogyakarta: Nuha Medika.
- Boestari, A. A., 2017. *Optimasi Komposisi Makanan Bagi Penderita Hipertensi Menggunakan Metode Particle Swarm Optimization*. Malang: Fakultas Ilmu Komputer Universitas Brawijaya.
- Humaniora, 2016. *Media Indonesia*. [Online] Tersedia di: < <http://www.mediaindonesia.com/news/read/83701/angka-kematian-ibu-masih-tinggi-1/2016-12-21>> [Diakses 15 September 2017].
- Indonesia, P. A. G., 2009. *Kamus Gizi*. Jakarta: PT Kompas Media Nusantara.
- Indriati, M. T. & Sukaca, B. E., 2015. *Nutrisi Janin & Bayi*. Yogyakarta: Parama Ilmu.
- Kaiser Lucia L. PhD, R., Davis & Christina G. Campbell, P. R., 2014. Nutrition and Lifestyle for a Healthy Pregnancy Outcome. *Journal of the Academy of Nutrition and Dietetics*.
- Kusumaningsih, F. D., 2016. *Penerapan Algoritma Genetika Pada Optimasi Susunan Bahan Makanan Untuk Pemenuhan Kebutuhan Gizi Keluarga*. Malang: Skripsi. Fakultas Ilmu Komputer Universitas. Brawijaya.
- Mahmudy, W. F., 2013. *Algoritma Evolusi*. Malang: Program Teknologi Informasi dan Ilmu Komputer.
- Mahmudy, W. F., 2015. *Dasar-Dasar Algoritma Evolusi*. Malang: Program Teknologi Informasi dan Ilmu Komputer.
- Maryamah, 2017. *Optimasi Komposisi Makanan Pada Penderita Diabetes Melitus dan Komplikasinya Menggunakan Algoritma Genetika*. Malang: Skripsi. Fakultas Ilmu Komputer Universitas. Brawijaya.
- Moehji, S., 2017. *Dasar-Dasar Ilmu Gizi 2*. Jakarta: Pustaka Kemang.
- Nurvenus, K., 2015. *Penerapan Algoritma Genetika untuk Optimasi Asupan Gizi Pasien Diet Khusus Dengan Biaya Minimal*. s.l.:s.n.
- Putri, R. R. A., 2018. *Optimasi Komposisi Menu Makanan Bagi Penderita Tekanan Darah Tinggi Menggunakan Algoritme Genetika Adaptif*. Malang: Skripsi. Fakultas Ilmu Komputer Universitas. Brawijaya.
- Sari, A. P., 2014. *Optimasi Asupan Gizi Ibu Hamil Dengan Menggunakan Algoritma Genetika*.
- Shafaat, M., 2017. *Optimasi Komposisi Makanan Diet Bagi Penderita Hipertensi menggunakan Algoritme Genetika*.
- Siahaan, E. J. I., 2017. *Sistem Rekomendasi Bahan Makanan Bagi Penderita Penyakit Jantung Menggunakan Algoritma Genetika*.

Wahid, N. & Mahmudy, W. F., 2015. Optimasi Komposisi Makanan Untuk Penderita Kolesterol Menggunakan Algoritma Genetika. *DORO:Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya*.

Waspadji, S., 2010. *Daftar Bahan Makanan Penukar (Klinis)*. III penyunt. Jakarta: Fakultas Kedokteran Universitas Indonesia.

